Topology Optimisation for Coupled Convection Problems

Alexandersen, Joe; Sigmund, Ole; Andreasen, Casper Schousboe; Aage, Niels

Publication date:
2013

Document Version
Publisher's PDF, also known as Version of record

Topology Optimisation for Coupled Convection Problems

Topologioptimering for koblede konvektionsproblemer

- Master Thesis (30 ECTS) -

With corrections: 20-03-2013

Author: Joe Alexandersen, s072713

Signature: _____

Supervisors: Professor Dr.Techn. Ole Sigmund

Assistant Professor Ph.D. Casper Schousboe Andreasen

Researcher Ph.D. Niels Aage

DTU MEK

Abstract

This thesis deals with topology optimisation for coupled convection problems. The aim is to extend and apply topology optimisation to steady-state conjugate heat transfer problems, where the heat conduction equation governs the heat transfer in a solid and is coupled to thermal transport in a surrounding fluid, governed by a convection-diffusion equation, where the convective velocity field is found from solving the isothermal incompressible steady-state Navier-Stokes equations. Topology optimisation is also applied to steady-state natural convection problems.

The modelling is done using stabilised finite elements, the formulation and implementation of which was done partly during a special course as prepatory work for this thesis. The formulation is extended with a Brinkman friction term in order to facilitate the topology optimisation of fluid flow and convective cooling problems. The derived finite element formulation is implemented in an object-oriented parallel finite element framework programmed in the *C++* programming language, developed by the TopOpt research group of the Department of Mechanical Engineering at The Technical University of Denmark.

The presented work is seen as contributing new research to the field of topology optimisation for multiphysics problems. The topology optimisation of conjugate heat transfer problems is not very well documented in the literature, with only a few notable papers on the subject and to the authors knowledge, topology optimisation has not yet been applied to natural convection problems. Although the presented results are very simple and remain academic, it is envisioned that by further development, the methodology presented in this thesis, can be used to optimise realistic industrial problems such as the cooling of combustion engines or electronics.

This thesis confines itself to steady-state laminar flow at low to moderate Reynolds, Péclet and Rayleigh numbers.

Resumé

Dette kandidatspeciale omhandler topologioptimering for koblede konvektionsproblemer. Målet er at udbygge og anvende topologioptimering til stationære varmetransmissions problemer, hvor varmeledningsligningen gælder i et givent fast materiale og er koblet til varmeoverførsel ved konvektion og diffusion i en omkringværende bevægende fluid. Naturlig konvektion bliver yderligere også behandlet.

Modelleringen opnåes ved brug af stabiliserede finite elements, hvilket er blevet formuleret og implementeret delvist i et specialkursus som forebyggende arbejde til dette speciale. Formuleringen er blevet udbygget med et Brinkman friktionsled for at muliggøre topologioptimering af problemer indenfor strøminger og konvektionskøling. Den udledte finite element formulering er blevet implementeret i en objekt-orienteret og parallel finite element kode skrevet i programmeringsproget *C++*. Koden er udviklet og vedligeholdt af TopOpt forskningsgruppen ved Institut for Mekanisk Teknologi, Danmark Tekniske Universitet.

Det fremlagte arbejde bidrager ny forskning til feltet indenfor topologioptimering af multifysiske problemer. Artikler omhandlende topologioptimering af koblede varmetransmissions problemer er mangelfulde, med få bemærkelsesværdige artikler om emnet. Yderligere er topologioptimering, såvidt vides, ikke før blevet anvendt til problemer med naturlig konvektion. Selvom de fremlagte resultater er relativt simple og af en akademisk karakter, så ses det som oplagt at den fremlagte metodik med yderligere udvikling vil kunne anvendes til at optimere realistike industrielle problemer, såsom køling af motorer og elektronik.

De behandlede problemer begrænser sig til stationære laminare tilstande ved lave til moderate Reynolds, Péclet og Rayleigh tal.

Foreword

This thesis, for the Master degree in 'Engineering Design and Applied Mechanics' (Mechanical Engineering), has been carried out following the specified guidelines at the Technical University of Denmark (DTU). The project supervisors have been Professor Dr.Techn. Ole Sigmund, Assistant Professor Ph.D. Casper Schousboe Andreasen and Researcher Ph.D. Niels Aage. The project counts for 30 ECTS points and has been completed during the 4th and final semester of the Master degree, over the period from the 3rd of September 2012 to the 22nd of February 2013.

I would like to thank Ole, Casper and Niels for their good advice and helpful guidelines, as well as for fruitful discussions. It has been great to have this trifecta of clever minds to help me when I got stuck; Ole giving the broad and specialised information on topology optimisation, Casper providing theoretical insight into fluid mechanics and Niels giving expert advice on numerical modelling and programming. I would also like to extend my gratitude to Boyan Lazarov for helping me with various C++ and cluster-related questions. My parents and brother; Gitte, Soren and Mike Alexandersen, are thanked for giving me an enjoyable winter season in the Canadian snow, with plenty of good food, good company and pleasant times. My girlfriend, Ida Petersen, deserves special thanks for supporting me and giving me the energy to push through it all, when debugging and stabilised finite elements were getting me down. Finally, my friends, office-mates and StudentHPC partners should be thanked for making the long days at the office enjoyable.

Contents

Chapter 1

Introduction

The aim of this thesis is to extend and apply topology optimisation to coupled convection problems. In the context of this thesis, the term 'coupled convection problems' covers two types of physical problems.

The first set of problems are steady-state conjugate heat transfer problems, where the heat conduction equation governs the heat transfer in a solid and is coupled to thermal transport in a surrounding fluid, governed by a convection-diffusion equation, where the convective velocity field is found from solving the isothermal incompressible steady-state Navier-Stokes equations. The fields are thus loosely coupled; that is, the velocity field affects the temperature field, but not vice versa.

The second set of problems investigated, are steady-state conjugate natural convection problems. By taking density variations due to temperature differences into account through the Boussinesq approximation, the temperature and velocity fields are tightly coupled; that is, there is a two-way coupling between the temperature and velocity fields.

This thesis confines itself to steady-state laminar flow at low to moderate Reynolds, Péclet and Rayleigh numbers.

1.1 Topology optimisation

Topology optimisation is a material distribution method used for finding an optimal structural layout, for a given problem subject to design constraints. It is based on the classical mathematical discipline of optimisation working together with numerical models, such as Finite Element Modelling [FEM], and is an incredibly powerful tool. It has its roots in classic shape and size optimisation, where topology optimisation solves both problems at once. The method will be described in more detail in section 4.1, but a brief introduction to the design parameterisation is given below.

The material distribution, or structural design, is parameterised by defining an elementwise constant design variable, γ_e. In the true nature of the

problem, this design variable is discrete and should either represent solid material or fluid, for instance:

$$\gamma_e = \{0, 1\} \tag{1.1}$$

where 0 represents solid material and 1 represents fluid. As discrete optimisation is incredibly difficult to solve for large numbers of design variables, which is often the case for topology optimisation problems where each finite element is attributed a design variable, the problem is relaxed by allowing continuous variables:

$$\gamma_e \in [0, 1] \tag{1.2}$$

This relaxation allows for the use of gradient-based continuous optimisation methods.

The material properties are then interpolated as a function of the design variable in order to provide a continuous transition between two materials or phases, in this case solid or fluid. To avoid areas of intermediate density and to push the continuous variable towards a solution with design variables of only the values 0 and 1 (from now on referred to as a 0-1 solution or design), penalisation is introduced and this will be covered in section 4.1

1.2 Motivation

The topology optimisation of heat transfer problems has so far been more or less concentrated on pure conductive heat transfer. Some work has been done to incorporate design-dependent convection heat transfer to a surrounding fluid by interpolating convection boundaries into the design domain and applying a constant convective heat transfer coefficient on the solid-void boundaries [5, 6, 17, 33]. This methodology introduces the dependence of convective heat transfer, from a solid to a surrounding fluid, on the surface area. However, by assuming a constant convective heat transfer coefficient, the dependence on the fluid flow around the solid body is neglected.

To achieve greater accuracy in the modelling of the physical situation, topology optimisation will be extended to simulations where the fluid dynamics of the surrounding fluid is taken into account. By modelling the full conjugate heat transfer problem, the dependence of the convective heat transfer on the shape and surface of the solid body will be automatically included.

The presented work is seen as contributing new research to the field of topology optimisation for multiphysics problems. The topology optimisation of conjugate heat transfer problems is not very well documented in the literature, with only a few notable papers on the subject [22, 38, 40, 66], and to the authors knowledge, topology optimisation has not yet been applied to natural convection problems. Although the presented results are very

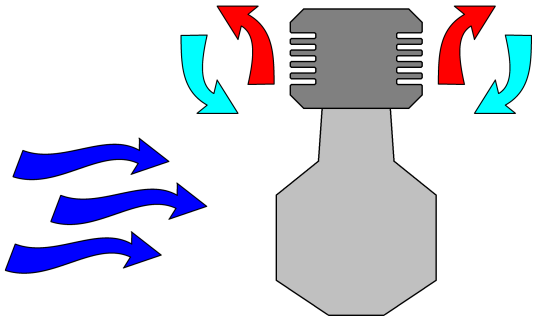


Figure 1.1 – Schematic illustration of the cooling of a combustion engine through forced and natural convection.

simple and remain academic, it is envisioned that by further development, the methodology presented in this thesis can be used to optimise realistic industrial problems such as the cooling of combustion engines as seen in figure 1.1.

Part I

Theory

Chapter 2

Fluid dynamics

This chapter covers the basics of fluid dynamics relevant to this thesis. The reader is refered to the many textbooks on the subject, e.g. [65], for more information about fluid dynamics in general.

2.1 Governing equations of fluid dynamics

Fluid dynamics is the description of the way a fluid flows. It is a complicated part of mechanical physics, due to the fact that many fluid flows exhibit nonlinear, multiscale and/or chaotic phenomena. A "solid" understanding of fluid dynamics is very important in many engineering applications, due to almost all structures and problems being exposed to interaction with a surrounding fluid to varying degree.

The three fundamental laws of mechanics are the conservation of momentum, mass and energy. They are presented below in their most general form for a moving fluid. The conservation of momentum in three-dimensions gives rise to three coupled nonlinear non-homogenous partial differential equations, known as the Navier-Stokes equations:

$$\rho\frac{\partial u_i}{\partial t} + \rho u_j\frac{\partial u_i}{\partial x_j} = s_i + \rho g_i + \frac{\partial \sigma_{ij}}{\partial x_j} \tag{2.1}$$

where ρ is the fluid density, u_i is the velocity vector, s_i is a volumetric momentum source term, g_i is the gravity vector and σ_{ij} is the fluid stress tensor defined as:

$$\sigma_{ij} = -\delta_{ij}p + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \delta_{ij}\lambda\frac{\partial u_k}{\partial x_k} \tag{2.2}$$

where p is the pressure, μ is the dynamic viscosity of the fluid, λ is the bulk viscosity and δ_{ij} is Kronecker's delta. The conservation of mass for a moving fluid gives rise to the continuity equation, a homogenous partial differential

equation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0 \tag{2.3}$$

Finally, the first law of thermodynamics gives rise to the conservation of energy or the energy equation:

$$\rho \frac{\partial h}{\partial t} + \rho u_j \frac{\partial h}{\partial x_j} = s_T + \frac{\partial p}{\partial t} + u_j \frac{\partial p}{\partial x_j} + \frac{\partial}{\partial x_j}\left(k \frac{\partial T}{\partial x_j}\right) + \sigma_{ij}^{'} \frac{\partial u_i}{\partial x_j} \tag{2.4}$$

where h is the enthalpy, s_T is the volumetric heat generation term, k is the thermal conductivity of the fluid, T is the temperature and $\sigma_{ij}^{'}$ is the viscous stress tensor defined as:

$$\sigma_{ij}^{'} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \delta_{ij}\lambda \frac{\partial u_k}{\partial x_k} \tag{2.5}$$

Assuming constant fluid properties, incompressible isothermal flow and neglecting viscous dissipation, equations (2.1) - (2.4) can be simplified to:

$$\rho \frac{\partial u_i}{\partial t} + \rho u_j \frac{\partial u_i}{\partial x_j} = s_i + \rho g_i + \frac{\partial \sigma_{ij}}{\partial x_j} \tag{2.6a}$$

$$\frac{\partial u_j}{\partial x_j} = 0 \tag{2.6b}$$

$$\rho c_p \frac{\partial T}{\partial t} + \rho c_p u_j \frac{\partial T}{\partial x_j} = s_T + k \frac{\partial T}{\partial x_j \partial x_j} \tag{2.6c}$$

where c_p is the specific heat capacity, under constant pressure, for the fluid. Further assuming steady-state conditions, which is the case throughout this thesis, the final set of equations governing the conservation of mass, momentum and energy for incompressible isothermal steady-state fluid flow become:

$$\rho u_j \frac{\partial u_i}{\partial x_j} - \frac{\partial \sigma_{ij}}{\partial x_j} = s_i + \rho g_i \tag{2.7a}$$

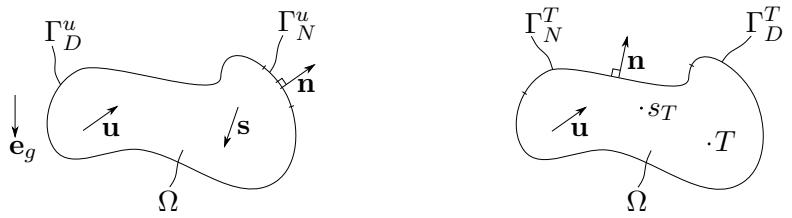
$$\frac{\partial u_j}{\partial x_j} = 0 \tag{2.7b}$$

$$\rho c_p u_j \frac{\partial T}{\partial x_j} - k \frac{\partial T}{\partial x_j \partial x_j} = s_T \tag{2.7c}$$

where the simplified fluid stress tensor is given by:

$$\sigma_{ij} = -\delta_{ij} p + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{2.8}$$

Equation (2.7a) describes the convection and diffusion of momentum, equation (2.7b) enforces a divergence free flow and equation (2.7c) describes the convection and diffusion of thermal energy quantified by the temperature.



(a) Momentum **(b)** Temperature

Figure 2.1 – Illustration of an arbitrary domain subject to boundary conditions, source terms and the governing equations.

The Navier-Stokes equations are very complex and only a few closed-form analytical solutions exist. Many of these analytical solutions are quite useful in certain application, however as the equations need to be simplfied significantly to be solvable analytically, this also imposes a limit on the complexity of the problems that can be solved. This is the reason as to why the Navier-Stokes equations are almost always solved numerically and why the field of computational fluid dynamics is an ever expanding area of research and application.

Figure 2.1 shows an arbitrary domain Ω subject to various boundary conditions and source terms. Figure 2.1a illustrates the boundary conditions for the momentum equations, as well as a source term $\mathbf{s}$, a gravitational acceleration in direction $\mathbf{e}_g$ and a convective velocity field $\mathbf{u}$. Γ_N^u is the subset of the outer surface on which Neumann condtions are applied, in the case of the Navier-Stokes equations this would be $\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{h}_n$ where a given stress is imposed on the boundary in the direction of the normal vector, $\mathbf{n}$. Γ_D^u is the subset of the outer surface on which Dirichlet condtions are applied, in the case of the Navier-Stokes equations this would be $\mathbf{u} = \mathbf{u}_D$ and/or $\mathbf{p} = \mathbf{p}_D$ where a given velocity vector and/or pressure is imposed on the boundary. Figure 2.1b illustrates the boundary conditions for the convection-diffusion equation, as well as source term s_T and a convective velocity field $\mathbf{u}$. Γ_N^T is the subset of the outer surface on which Neumann condtions are applied, in the case of the temperature equation this would be $\boldsymbol{\nabla} T \cdot \mathbf{n} = f_n$ where a given temperature flux is imposed on the boundary in the direction of the normal vector, $\mathbf{n}$. Γ_D^T is the subset of the outer surface on which Dirichlet conditions are applied, in the case of the temperature equation this would be $T = T_D$ where a given temperature is imposed on the boundary. The two physical problems are depicted in seperate subfigures in order to illustrate the fact that in general the Neumann and Dirichlet boundaries are not neccessarily the same for the two sets of governing equations.

2.2 The Boussinesq approximation

To include buoyancy effects due to temperature differences in the fluid, it is assumed that the volumetric gravity force can be written as:

$$\rho g_i = \rho_0 g_i + (\rho - \rho_0) g_i \tag{2.9}$$

where ρ_0 is the average fluid density. The fluid density is posed as a function of temperature and is Taylor-expanded about the defined average density:

$$\rho(T) \approx \rho_0 + \left(\frac{\partial \rho}{\partial T}\right)_{T=T_0} (T - T_0) \tag{2.10}$$

where T_0 is the temperature corresponding to ρ_0 and only the first term of the Taylor expansion has been used. By assuming only small temperature differences, the gradient of the density with respect to temperature is approximated as being linear:

$$\left(\frac{\partial \rho}{\partial T}\right)_{T=T_0} = -\rho_0 \beta \tag{2.11}$$

where β is the coeffecient of thermal volume expansion. This is inserted into equation (2.10) giving:

$$\rho(T) \approx \rho_0 - \rho_0 \beta \, (T - T_0) \tag{2.12}$$

Inserting equation (2.12) into equation (2.9) gives the Boussinesq approximation:

$$\rho g_i \approx \rho_0 g_i (1 - \beta \, (T - T_0)) \tag{2.13}$$

Inserting equation (2.13) into equation (2.7) yields the set of equations governing the conservation of mass, momentum and energy for incompressible steady-state fluid flow taking bouyancy effects into account:

$$\rho_0 u_j \frac{\partial u_i}{\partial x_j} - \frac{\partial \sigma_{ij}}{\partial x_j} = s_i + \rho_0 \, g_i \, \beta (T - T_0) \tag{2.14a}$$

$$\frac{\partial u_j}{\partial x_j} = 0 \tag{2.14b}$$

$$\rho_0 c_p u_j \frac{\partial T}{\partial x_j} - k \frac{\partial T}{\partial x_j \partial x_j} = s_T \tag{2.14c}$$

where the gravitational body force has been absorbed into the pressure, by using the fact that gravity is a conservative force and it therefore can be represented as the gradient of a scalar quantity, modifying the pressure to include the so-called "gravitational head" - see [7, Section 2.3.4] and appendix A.2 for further explanation.

Name	Definition	Description
Reynolds	$Re = \frac{UL\rho}{\mu} = \frac{UL}{\nu}$	$\frac{\text{intertial forces}}{\text{viscous forces}}$
Péclet	$Pe = \frac{ULk}{\rho c_p} = \frac{UL}{\Gamma}$	$\frac{\text{convective flux}}{\text{diffusive flux}}$
Prandtl	$Pr = \frac{Pe}{Re} = \frac{\nu}{\Gamma}$	$\frac{\text{momentum diffusitivy}}{\text{thermal diffusivity}}$
Grashof	$Gr = \frac{g\beta\Delta TL^3}{\nu^2}$	$\frac{\text{buoyancy forces}}{\text{viscous forces}}$
Rayleigh	$Ra = GrPr = \frac{g\beta\Delta TL^3}{\nu\Gamma}$	often used parameter
Richardson	$Ri = \frac{Gr}{Re^2} = \frac{g\beta\Delta TL^3}{\Gamma^2}$	$\frac{\text{potential energy}}{\text{kinetic energy}}$

Table 2.1 – Dimensionless numbers

2.3 Dimensionless form

The dimensionless form of equations (2.14) is used as a basis for the finite element formulation and is as follows:

$$u_j \frac{\partial u_i}{\partial x_j} - \frac{\partial \sigma_{ij}}{\partial x_j} = s_i + b_i \tag{2.15a}$$

$$\frac{\partial u_j}{\partial x_j} = 0 \tag{2.15b}$$

$$u_j \frac{\partial T}{\partial x_j} - \frac{1}{Pe} \frac{\partial T}{\partial x_j \partial x_j} = s_T \tag{2.15c}$$

where the dimensionless stress tensor is given by:

$$\sigma_{ij} = -\delta_{ij} p + \frac{1}{Re}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{2.16}$$

and the Boussinesq forcing term is given by:

$$b_i = \frac{Ra}{RePe} e_i^g T = Ri\, e_i^g T \tag{2.17}$$

A detailed derivation of the dimensionless equations can be found in [7].

There are several dimensionless numbers which appear in the nondimensionalisation process and many of them can be interrelated several ways as can be seen in table 2.1.

The Reynolds number, Re, represents the ratio between intertial and viscous forces and thus describes to what extent the flow is diffusion- or convection-dominated. The Reynolds number is used to determine whether the flow is in the laminar, transition or turbulent regime. At the limit of $Re \to 0$ one obtains Stokes flow and as the Reynolds number increases one generally obtains turbulent flow.

The Péclet number, Pe, represents the ratio between the convective and diffusive heat transfer and thus whether the heat transfer is diffusion- or convection dominated.

The Prandtl number, Pr, is a fluid material constant and describes the ratio between the momentum and thermal diffusivities of the fluid. For low Pr heat conduction is effective compared to convection and for high Pr convection is more effective for transfering energy through the fluid.

The Grashof number, Gr, represents the ratio between buoyancy and viscous forces. The number is frequently used for natural convection flows and can be used to determine the onset of turbulent natural convection.

The Rayleigh number, Ra, can be seen as the natural convection equivalent of the Péclet number and describes to what extent the heat transfer is diffusion- or convection-dominated.

The Richardson number, Ri, represent the ratio between the potential and kinetic energy of the fluid system. The Richardson number is used to determine whether a flow is dominated by forced or natural convection as will be described in the following section.

2.4 Forced, mixed or natural convection

There are three main scenarios for the type of convection taking place and these are described below with reference to the Richardson number:

1. $Ri << 1$: Forced convection dominates
2. $Ri >> 1$: Natural convection dominates
3. $Ri \approx \mathcal{O}(1)$: Mixed natural and forced convection

In the extremity of case 1, the buoyancy effects are so small compared to the interial effects, that the Boussinesq forcing term can be neglected. The Richardson number is set to zero and the velocity and temperature fields are thus decoupled, recovering the original steady-state isothermal incompressible flow equations in equation (2.7).

In the extremity of case 2, where no forcing velocity is present at all, the reference velocity used to non-dimensionalise the flow equations is defined as the diffusion-velocity:

$$U_{\text{diff}} = \frac{\Gamma}{L} \tag{2.18}$$

This in turn reduces the Péclet number to unity, $Pe = 1$, and the Reynolds number to the reciprocal of the Prandtl number, $Re = \frac{\Gamma}{\nu} = \frac{1}{Pr}$. Therefore, the non-dimensional coefficient in equation (2.17) reduces to the following for pure natural convection problems:

$$\frac{Ra}{RePe} = GrPr^2 = Ri_{\text{n}} \tag{2.19}$$

which is the Richardson number for pure natural convection problems.

In case 3, neither natural nor forced convection effects can be neglected and mixed convection is said to be taking place.

2.5 Brinkman friction term

The flow through an idealised porous medium can be modelled by adding a velocity-dependent spatially-varying Brinkman-friction term, αu_i, to the momentum equations:

$$u_j \frac{\partial u_i}{\partial x_j} - \frac{1}{Re} \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \frac{\partial p}{\partial x_i} + \alpha(\mathbf{x})\, u_i = s_i \tag{2.20}$$

where α is the inverse permeability of the porous medium.

In topology optimisation for fluid flow problems, where the goal is to enforce zero-velocity in the solid domain and no-slip conditions along the interfaces, the Brinkman term is mainly seen as an algorithmic device driving velocities to zero inside the solid. Here, the Brinkman term can be seen as a penalisation of the velocities in the parts of the domain where α obtains a large positive value. Mathematically one would like to make use of an infinitely impermeable solid, however, in practice to get numerically stable solutions, one must limit the upper bound to a "suitably large number".

Note on the scaling of the Brinkman coefficient

Although the penalisation parameter, α, in topology optimisation usually assumes the role of a purely algorithmic way to enfore zero velocities in the solid parts of the domain and thereby no-slip conditions along the interface, it is seen as beneficial to look at the physical scaling of the parameter. The dimensional Navier-Stokes with a Brinkman porosity term added yields:

$$\rho u_j \frac{\partial u_i}{\partial x_j} - \mu \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \frac{\partial p}{\partial x_i} + \frac{\mu}{\kappa} u_i = s_i \tag{2.21}$$

where μ is the dynamic viscosity and κ the permeability of the porous medium. By non-dimensionalising equation (2.21), the dimensionless Navier-Stokes equations with Brinkman penalty term become:

$$u_j \frac{\partial u_i}{\partial x_j} - \frac{1}{Re} \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \frac{\partial p}{\partial x_i} + \alpha u_i = s_i \tag{2.22}$$

where the the penalisation factor is given by:

$$\alpha = \frac{1}{ReDa} \tag{2.23}$$

Re is the Reynolds number and Da is the Darcy number, which is defined as the dimensionless permeability:

$$Da = \frac{\kappa}{L^2} \tag{2.24}$$

From this physical point of view, it can be seen that the penalisation factor should be inversely proportional to the Reynolds number. A paper by

Khadra et al. [34] on the fictitious domain approach for the numerical solution of incompressible viscous flows with immersed objects, uses the same scaling to obtain a penalisation factor that is dependent on the Reynolds number. From a numerical point of view, it has been observed that for the same problems, a much larger penalisation factor is needed when the Reynolds number is low as compared to when the Reynolds number is large. Furthermore, it has been observed that using the same high penalisation factor needed for a low Reynolds number, for the same problem with a larger Reynolds number leads to numerical difficulties and poor convergence of the Newton solver. It is therefore argued that the upper limit of the penalisation factor, α, under topology optimisation of fluid flows should be inversely proportional to the Reynolds number. This is further investigated in section H.1.

Chapter 3

Modelling

3.1 The finite element method for fluid dynamics

The Galerkin finite element method was introduced to the field of fluid mechanics in the 1970s. Due to the method's extreme success within the field of solid mechanics and solid heat transfer, the method was thought to revolutionise the field of computational fluid dynamics. However, this was not the case, initially, as the standard Galerkin finite element method has difficulties coping with convection-dominated problems and often produces oscillatory solutions. These spurious node-to-node oscillations, caused by large solution gradients, could initially only be removed by extreme mesh refinement. But with the introduction of stabilisation techniques in the 1980s, the finite element method overcame its initial problems and has proved to be a strong contender in the computational modelling of fluid dynamics. A notable paper on the Galerkin finite element method in fluids is the 1973 paper by Taylor and Hood [56], who introduced the mixed finite element formulation utilising the velocity and pressure as field variables. An overview of the finite element method for fluid flows can be found in the books by Taylor and Hughes [57], Donea and Huerta [26] and Zienkiewicz and Taylor [70].

The finite element formulation used throughout this thesis, was derived, implemented and tested by the author during a special course, as preparatory work for this thesis. A brief overview of the final matrix equations will be given in the following subsections, but the reader is refered to [7] for further detailed information. In order for this thesis to stand as a self-contained report, the element matrices and vectors are described in appendix A.

3.2 Finite element equations

3.2.1 Variational weak form

To obtain the finite element discretised equations, the weak form of the governing equations is found by multiplying the strong form, equations (2.15), with suitable test functions and integrating over the domain. A detailed derivation is given in [7]. The suitable finite dimensional spaces, $\boldsymbol{\mathcal{S}}^h$, $\mathcal{S}^h$, $\boldsymbol{\mathcal{V}}^h$, $\mathcal{V}^h$ and $\mathcal{Q}^h$, are introduced and the discrete variational problem becomes:

Find $\mathbf{u}^h \in \boldsymbol{\mathcal{S}}^{\boldsymbol{h}}$, $p^h \in \mathcal{Q}^h$ and $T^h \in \mathcal{S}^h$ such that $\forall\, \mathbf{w}^h \in \boldsymbol{\mathcal{V}}^{\boldsymbol{h}}$, $\forall\, q^h \in \mathcal{Q}^h$ and $\forall\, w^h \in \mathcal{V}^h$:

$$\int_\Omega \overbrace{\mathbf{w}^h \cdot \Big(\mathbf{u}^h \cdot \boldsymbol{\nabla}\Big)\, \mathbf{u}^h}^{\text{convection}}\, dV + \int_\Omega \overbrace{\boldsymbol{\nabla}\mathbf{w}^h : \frac{1}{Re}\Big(\boldsymbol{\nabla}\mathbf{u}^h + \Big(\boldsymbol{\nabla}\mathbf{u}^h\Big)^{\mathrm{T}}\Big)}^{\text{diffusion}}\, dV - \int_\Omega \overbrace{\Big(\mathbf{w}^h \cdot \boldsymbol{\nabla}\Big)\, p^h}^{\text{pressure coupling}}\, dV$$
$$- \int_\Omega \underbrace{\mathbf{w}^h \cdot \mathbf{s}}_{\text{body force}}\, dV - \int_\Omega \underbrace{\mathbf{w}^h \cdot \mathbf{b}\Big(T^h\Big)}_{\text{Boussinesq force}}\, dV - \int_{\Gamma_N^u} \underbrace{\mathbf{w}^h \cdot \mathbf{h}_n}_{\text{traction}}\, dS = 0 \tag{3.1a}$$

$$- \int_\Omega \underbrace{q^h\, \boldsymbol{\nabla} \cdot \mathbf{u}^h}_{\text{continuity}}\, dV = 0 \tag{3.1b}$$

$$\int_\Omega \overbrace{w\, \mathbf{u}^h \cdot \boldsymbol{\nabla} T^h}^{\text{convection}}\, dV + \int_\Omega \overbrace{\frac{1}{Pe}\boldsymbol{\nabla} w^h \cdot \boldsymbol{\nabla} T^h}^{\text{diffusion}}\, dV$$
$$- \int_\Omega \underbrace{w^h\, s_T}_{\text{body force}}\, dV - \int_{\Gamma_N^T} \underbrace{\frac{1}{Pe} w^h\, f_n}_{\text{flux}}\, dS = 0 \tag{3.1c}$$

The Brinkman friction term is included in the weak form by setting the source term to $\mathbf{s} = -\alpha\mathbf{u}$.

3.2.2 Matrix equations

The standard Galerkin finite element method results in the following discretised system of equations, in its most general form including the Boussinesq coupling and Brinkman term:

$$\begin{bmatrix} \mathbf{C}(\mathbf{u}) + \mathbf{K} + \boldsymbol{\alpha} & \mathbf{G} & \mathbf{B} \\ \mathbf{G}^{\mathrm{T}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_t(\mathbf{u}) + \mathbf{K}_t \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{p} \\ \mathbf{t} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} \\ \mathbf{0} \\ \mathbf{f}_t \end{Bmatrix} \tag{3.2}$$

where $\mathbf{u}$ is the vector containing the velocity degrees of freedom, $\mathbf{p}$ is the vector containing the pressure degrees of freedom, $\mathbf{t}$ is the vector containing the temperature degrees of freedom, $\mathbf{C}(\mathbf{u})$ is the nonlinear and nonsymmetric convection matrix, $\mathbf{K}$ is the diffusivity/viscosity matrix, $\boldsymbol{\alpha}$ is the

Brinkman matrix, $\mathbf{B}$ is the Boussinesq coupling matrix, $\mathbf{C}_t(\mathbf{u})$ is the non-symmetric temperature convection matrix, $\mathbf{K}_t$ is the temperature diffusivity matrix, $\mathbf{f}$ is the momentum forcing vector and $\mathbf{f}_t$ is the heat flux vector. The global state system of equations can be seen to be both nonlinear and nonsymmetric, both leading to difficulties for solution methods.

The global vectors and matrices are all obtained from the finite element assembly of all the element equivalents:

$$\square = \mathop{\mathsf{A}}_{e=1}^{n_\mathrm{e}} \square^e \tag{3.3}$$

where $\mathsf{A}_{e=1}^{n_\mathrm{e}}$ is the finite element assembly operator. The element level vectors and matrices, $\square^e$, are defined in appendix A and [7, Chapter 2].

3.2.3 Stabilised matrix equations

The Pressure-Stabilising/Petrov-Galerkin (PSPG) stabilisation method is used in order to allow for the use of equal-order interpolation for the velocity and pressure fields. The PSPG stabilisation was first introduced for the Stokes equations by Hughes et al. [32] and later generalised for the incompressible Navier-Stokes equations by Tezduyar et al. [62] and has since seen widespread use in the finite element modelling of fluid flow. The PSPG stabilisation affects the discrete continuity equation and allows otherwise unstable elements to circumvent the Ladyzhenskaya–Babuska-Brezzi (LBB), or inf-sup, stability condition for the finite element spaces. This condition is satisfied by a range of different combinations of finite element spaces, such as second-order and first-order interpolation for the velocity and pressure fields, respectively. But using higher-order elements for topology optimisation quickly becomes computationally expensive, as here one ideally wants to refine the mesh quite heavily in order to capture the design with a high resolution.

Furthermore, the Streamline-Upwind/Petrov-Galerkin (SUPG) stabilisation method, as presented by Brooks and Hughes [16], is used in order to supress oscillations in the velocity and temperature fields due to sharp solution gradients in the streamline direction, which often arise in convection-dominated problems due to downstream boundary conditions. The SUPG stabilisation method can be seen as a generalisation of upwinding schemes in finite difference and volume methods, adding a carefully scaled amount of numerical diffusion in the streamline direction.

The PSPG- and SUPG-stabilised Galerkin finite element method results in the following discretised system of equations, in its most general form

including the Boussinesq coupling and Brinkman term:

$$\boldsymbol{\mathscr{R}}(\mathbf{u},\mathbf{p},\mathbf{t}) = \begin{bmatrix} \mathbf{C}(\mathbf{u}) + \mathbf{K} + \boldsymbol{\alpha} + \mathbf{L}_\delta(\mathbf{u}) + \mathbf{L}_{\alpha\delta}(\mathbf{u}) & \mathbf{G} + \mathbf{A}_\delta(\mathbf{u}) & \mathbf{B} + \mathbf{Q}_\delta(\mathbf{u}) \\ \mathbf{G}^T + \mathbf{L}_\epsilon + \mathbf{L}_{\alpha\epsilon} & \mathbf{A}_\epsilon & \mathbf{Q}_\epsilon \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_t(\mathbf{u}) + \mathbf{K}_t + \mathbf{Q}_\zeta(\mathbf{u}) \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{p} \\ \mathbf{t} \end{Bmatrix} - \begin{Bmatrix} \mathbf{f} + \mathbf{r}_\delta(\mathbf{u}) \\ \mathbf{r}_\epsilon \\ \mathbf{f}_t + \mathbf{r}_\zeta(\mathbf{u}) \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix} \tag{3.4}$$

where $\boldsymbol{\mathscr{R}}$ denotes the residual of the equation system, subscript ϵ denotes PSPG stabilisation, subscript δ denotes SUPG stabilisation, subscript ζ denotes thermal SUPG stabilisation and all of the matrices and vectors are as defined in appendix A and [7, Section 2.5]. It should be noted, that only the direct dependence on the velocity field is noted above. The implicit dependence of the stabilisation parameters on the velocity and temperature fields has been left out for simplicity, but the full nonlinear dependence is shown in appendix B.

When considering problems, where the temperature field does not affect the velocity field, the flow and transport problems can be solved in a segregated manner:

$$\boldsymbol{\mathscr{R}}(\mathbf{u},\mathbf{p}) = \begin{bmatrix} \mathbf{C}(\mathbf{u}) + \mathbf{K} + \boldsymbol{\alpha} + \mathbf{L}_\delta(\mathbf{u}) + \mathbf{L}_{\alpha\delta}(\mathbf{u}) & \mathbf{G} + \mathbf{A}_\delta(\mathbf{u}) \\ \mathbf{G}^{\mathrm{T}} + \mathbf{L}_\epsilon + \mathbf{L}_{\alpha\epsilon} & \mathbf{A}_\epsilon \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{p} \end{Bmatrix} - \begin{Bmatrix} \mathbf{f} + \mathbf{r}_\delta(\mathbf{u}) \\ \mathbf{r}_\epsilon(\mathbf{u}) \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \end{Bmatrix} \tag{3.5a}$$

$$\left(\mathbf{C}_t(\mathbf{u}) + \mathbf{K}_t + \mathbf{Q}_\zeta(\mathbf{u})\right)\mathbf{t} = \mathbf{f}_t + \mathbf{r}_\zeta(\mathbf{u}) \tag{3.5b}$$

where the flow field is solved for first and subsequently the temperature field is found using the flow field as the advection field.

3.3 Newton's method

Newton's method can be used to solve systems of nonlinear equations, such as the equation systems generated by both the stabilised and unstabilised finite element formulations of the Navier-Stokes equations. The equation system is linearised such that a step is calculated from the following system of linear equations:

$$\left.\frac{d\boldsymbol{\mathscr{R}}}{d\mathbf{s}}\right|_k \boldsymbol{\Delta}\mathbf{s}_{k+1} = -\boldsymbol{\mathscr{R}}(\mathbf{s}_k) \tag{3.6}$$

where $\mathbf{s}$ is the vector of unknowns ($\mathbf{u}$, $\mathbf{p}$ and/or $\mathbf{t}$), $\boldsymbol{\Delta}\mathbf{s}$ is the Newton step, the subscripts denote iteration number, $\left.\frac{d\boldsymbol{\mathscr{R}}}{d\mathbf{s}}\right|_k$ is the Jacobian matrix, or tangent stiffness matrix, for the nonlinear system of equations at the k'th

iteration, and $\mathscr{R}(\mathbf{s}_k)$ is the residual of the nonlinear system of equations at the k'th iteration, as defined by either equation 3.4 or equation 3.5a.

The implemented nonlinear solver utilises the above linearisation and updates the state variables using:

$$\mathbf{s}_{k+1} = \mathbf{s}_k + \eta \boldsymbol{\Delta} \mathbf{s}_{k+1} \tag{3.7}$$

where $\eta \in (0, 1]$ is a user-specified steplength parameter. For many of the problems studied during this thesis, the full Newton step, $\eta = 1$, could be taken, leading to very fast convergence. However, as the nonlinearity of the problem increases, for instance due to a high *Re*- or *Ra*-number, smaller steps need to be taken in order to converge. Adaptive schemes have been tested for updating the steplength parameter during the nonlinear iterations, but these have been very ineffective for the tested problems. This will be discussed further in section 7.2.

For Newton's method to converge, a good start guess must be supplied. For the stabilised FEM used in this thesis, it has proven very effective to use a pseudo-stabilised flow solution as the initial guess. This initial guess comes from solving the flow equations for constant $\tau_{PS} = \tau_{SU} = 0.01$ and $\mathbf{u} = \mathbf{0}$ and is thus a smooth and over-stabilised Stokes flow solution.

The stabilised system of equations is a quite complex and nonlinear system and has proven quite difficult to derive the analytical tangent stiffness matrix for. Currently, the contributions to the tangent state matrix from the differentiated state stiffness matrix and right-hand side vector are calculated in MAPLE. The various element matrices and vectors are differentiated by looping over all element state variables and assemblying the matrix contributions as described in appendix B.

3.4 Short on DFEM

DFEM is an object-oriented parallel finite element framework programmed in the *C++* programming language [52]. DFEM is used and developed by the TopOpt research group of the Department of Mechanical Engineering at The Technical University of Denmark. It has been developed to provide a user-friendly framework in which researchers can programme their given finite element formulations and carry out largescale parallel computations [4]. The basic thought behind DFEM is that all one needs to do in order to introduce a new finite element formulation, that does not already exist in the element library, is to create an element class which basically sets up the degrees of freedom and builds the element matrices and vectors. All of the geometry, element connectivity, parallel message passing and so on, is safely hidden in DFEM's many libraries, so that the user does not need to be concerned about that. To utilise ones newly added finite element formulation, a driver needs to be programmed. A driver is basically a set

of instructions to call pre-existing functions that sets up the model, the underlying equation system and solves it.

The finite element formulation laid out in appendix A has been implemented into DFEM by the use of MAPLE scripts. That is, MAPLE scripts have been used to analytically derive the expressions for the element matrices and vectors, which are then exported to *C++* format and implemented into DFEM Navier-Stokes elements, within which the numerical integration of the matrices and vectors take place using standard Gaussian quadrature, as described in e.g. [21, Chapter 6].

3.5 Direct solver

Due to the system matrices, for both the Navier-Stokes equations and the temperature equation, being non-symmetric and furthermore indefinite for the Navier-Stokes equations, a direct solution method is used to solve the equation system at each nonlinear iteration. The direct solver package MUMPS [8] is used to LU-factorise the system matrix and solve the system. MUMPS is a multifrontal parallel direct solver which is very popular and efficient for largescale applications.

Many of the examples in Part II and appendix H of this thesis are two-dimensional problems and the approach of utilising a direct solver has been very efficient. However, this approach quickly becomes uneconomical when dealing with three-dimensional problems and iterative solvers for general unsymmetric matrices, such as GMRES, need to be explored to deal with this. This is further discussed in section 7.2.

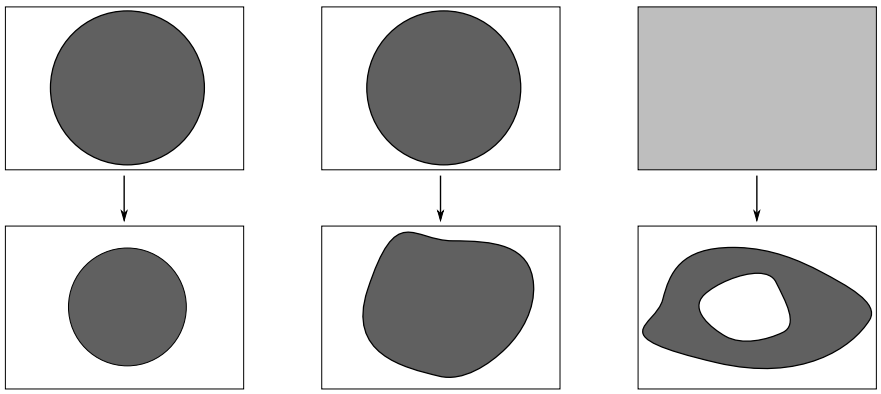
Chapter 4

Topology optimisation

This chapter begins with an introduction to general topology optimisation. Thereafter, topology optimisation for fluid flow problems is described and the derivations and considerations neccessary for the topology optimisation of fluid-solid thermal problems are presented. Finally, the metodology for solving the topology optimisation problems is described, hereunder the derivation of adjoint sensitivities. For a brief overview of general optimisation theory, the reader is refered to appendix C.

4.1 General topology optimisation

Figure 4.1 illustrates the three standard methods for structural optimisation; sizing, shape and topology. Sizing optimisation is probably the most classical of the three methods and still widely used by engineers, for instance when dimensioning a beam of a given cross-section against buckling or a bolt



(a) Sizing **(b)** Shape **(c)** Topology

Figure 4.1 – Conceptual illustration of the three standard methods for structural optimisation.

against yielding. Thus for sizing optimisation, the shape is defined apriori and the size is optimised. Shape optimisation is a method where the original shape of an object is changed in order to optimise a given function. Shape optimisation is often limited to small adjustments of the shape, as large distortions can lead to problems with the underlying discretisation. Thus, for shape opimisation the approximate shape is defined apriori and adjusted by the optimisation. Common to both sizing and shape optimisation is that an initial shape has to be defined apriori, which limits the design freedom. Topology optimisation is a method where both the shape, size and topology is optimised simultaneously. Topology can be seen as a generalisation of shape and describes the properties that are conserved through deformation, twisting, and stretching of objects. The initial circular shape in figures 4.1a and 4.1b have the same topology as their respective end results, as each can be seen as a deformation and stretching of the inital object. However, the end result from the topology optimisation, figure 4.1c, has a different topology to the other shapes due to the hole. Thus, a simplified explanation of topology optimisation is the optimisation of where to put the holes. The initial domain in figure 4.1c is coloured grey to indicate that topology optimisation does not need an initial guess of the shape.

Topology optimisation as it is known today was pioneered by Bendsøe and Kikuchi [12] as a material distribution method used for finding an optimal structural layout, for a given problem subject to design constraints. The most popular numerical method for topology optimisation, now known as the density or SIMP approach, was developed concurrently to the homogenisation approach founded by Bendsøe and Kikuchi [12]. The SIMP approach was originally suggested by Bendsøe [11] and used extensively by Rozvany et al. [47], Zhou and Rozvany [68], who also suggested the term 'solid isotropic microstructure with penalisation' (SIMP). The term 'density approach' is more general and used to denote the topology optimisation method where the topology and material properties are interpolated based on a relative density variable used as the design variable. Although the topology optimisation method originated and gained maturity within structural mechanics, the method has since been extended to a wide range of physics, such as acoustics [28], photonics [64], fluidics [14], thermal problems [13] and many more.

In the continuous setting, the goal of topology optimisation is to determine whether there should be material present or not, at any given point in a domain. In practice, the continuous material distribution is discretised and the design is parameterised either by an element-wise constant design variable or nodal design variables. Throughout this thesis, the first approach is taken for each finite element.

For further general introduction to topology optimisation, the reader is refered to the monograph by Bendsøe and Sigmund [13] and the review article by Rozvany [46]. For more detailed information, the vast amount of

papers on topology optimisation should be consulted.

4.2 Fluid topology optimisation

Topology optimisation for fluid flow problems was pioneered for Stokes flow by Borrvall and Petersson [14]. They achieved control of the topology of a solid domain in Stokes flow, by the introduction of a friction term yielding the generalised Stokes equations. This friction term is what has become known as Brinkman penalisation. Borrvall and Petersson [14] used lubrication theory to relate the penalisation factor to the flow of a fluid through channels of varying heights in order to validate their model. Conceptually this means that for two-dimensional problems, the flow is seen as being contained between two lateral walls with varying distance between them. The topology optimisation for fluid flow problems was later extended to the Navier-Stokes equations by Gersborg-Hansen et al. [29] and Olesen et al. [44]. Gersborg-Hansen et al. [29] used the same lubrication theory approach as Borrvall and Petersson [14], whereas Olesen et al. [44] used a fictitious porous Brinkman medium. Brinkman penalisation had previously been used to model immersed objects and boundaries in fluid dynamics on fixed grids, but Borrvall and Petersson [14] were the first to apply this technique to the topology optimisation of fluid problems.

It is important to note that the lubrication theory and porous medium approaches are equivalent and interchangeable in three-dimensions, but that the two-dimensional lubrication theory approach requires a minimum penalisation coefficient in order to remain valid. By allowing the Brinkman penalisation coefficient to go to zero in two-dimensions, one obtains the uninhibited plane Navier-Stokes equations which is equivalent to having an infinite domain in the out-of-plane direction. In three-dimensions, the only logical option is to let the minimum penalisation be zero for both approaches, as the three-dimensionality of the modelling automatically takes the friction forces arising for flows between plates into account.

The Brinkman approach has since been used for transport problems [9], reactive flows [43], transient flows [23, 37], fluid-structure interaction [67], flows driven by body forces [24] and to a limited extent fluid-structure thermal transport problems, as will be discussed in section 4.2.2.

There exists alternatives to Brinkman penalisation in the literature. Guest and Prevost [31] utilised the interpolation between two physical models, namely the Darcy and Stokes equations. The levelset approach to topology optimisation has also been applied to fluid flow problems [18, 69] and recently, the level set approach was combined with the extended finite element method (X-FEM) by Kreissl and Maute [36]. The levelset approach to fluid flow topology optimisation is interesting, as many of the implementations only model the fluid flow problem in the domain defined to be

fluid by the level set function [18, 27, 36, 69]. Even though this generally will decrease the CPU time and memory spent on solving the state problem, the increased effort associated with remeshing and book-keeping throughout the optimisation process, has kept the method from becoming the prefered approach to fluid flow topology optimisation so far.

The Brinkman, or material distribution, approach to topology optimisation for fluid flow problems is not without further difficulties. Lee [38] shows design oscillations near the fluid-solid interfaces due to oscillations in the state and sensitivity fields and Kreissl and Maute [36] highlight the fact that pressure diffuses through the porous material, which at times causes unphysical flow solutions.

4.2.1 Interpolation of porosity

The standard within the Brinkman approach has been to use the convex interpolation function as laid out originally by Borrvall and Petersson [14] for the penalisation factor:

$$\alpha_{BP}(\gamma) = \overline{\alpha} + \gamma(\underline{\alpha} - \overline{\alpha}) \frac{1 + q_{BP}}{\gamma + q_{BP}} \tag{4.1}$$

for $q_{BP} > 0$ and $0 \leq \gamma \leq 1$, where decreasing q_{BP} increases the curvature of the interpolation function and the limit $q_{BP} \to \infty$ yields a linear interpolation. In this thesis, the RAMP interpolation function [51] is used instead:

$$\alpha(\gamma) = \underline{\alpha} + (\overline{\alpha} - \underline{\alpha}) \frac{1 - \gamma}{1 + \gamma q_\alpha} \tag{4.2}$$

for $q_\alpha \geq 0$ and $0 \leq \gamma \leq 1$. The difference between the two is very small, as the Borrvall-Petersson and RAMP functions are mathematically identical for $q_{BP} = \frac{1}{q_\alpha}$, except for the fact that the linear case is included in the latter when $q_\alpha = 0$ and not just as a limit case. Mainly due to this fact, as well as the RAMP function seeming more logical (increasing q_α, increasing curvature), the RAMP interpolation function of equation (4.2) has been used throughout this thesis. The interpolation function is shown in figure 4.2 for various values of the penalty factor.

Throughout this thesis, $\gamma_e = 0$ will represent solid and $\gamma_e = 1$ will represent fluid.

4.2.2 Interpolation of Péclet number

The difference in thermal conductivity in the solid and fluid domains is included through the interpolation of the Péclet number. Varying the thermal diffusivity and/or conductivity has, to the authors knowledge, only been dealt with in very few papers, as most publications focus on the temperature

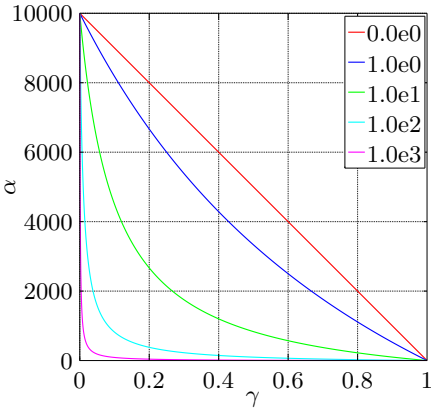


Figure 4.2 – Plot of the porosity interpolation function, equation (4.2), for $\underline{\alpha} = 0$, $\overline{\alpha} = 10^4$ and various values of the penalty factor, q_α.

distribution of the fluid and not the solid itself. There are a few notable exceptions, which are laid out in the following. Yoon [66] interpolates the conductivity and other parameters using SIMP functions in order to design heat dissipating structures subjected to forced convection. Matsumori et al. [40] presented results for a heat exchanger problem with a linearly-interpolated design-dependent volumetric heat generation. Dede [22] presented results for jet impingement surface cooling problems using linear interpolation for the thermal conductivity. Lee [38] interpolates the conductivity using the RAMP function and presents many interesting results for the design of convective cooling systems. The objective function used throughout [38] is the average temperature of the entire domain and seems to give rise to many difficulties involving intermediate design variables and unphysical flow solutions. Kontoleontos et al. [35] forces the temperature in the solid domain towards a prescribed wall temperature using the same penalisation technique as used to drive the velocities to zero. Lastly, McConnell and Pingen [41] interpolates the thermal diffusivity using a SIMP function for the design of layered pseudo-3D problems using the Lattice-Boltzmann-Method.

An investigation of candidates for the objective functional for topology optimisation problems investigated in this thesis is documented in appendix H.

Criteria for interpolation function

The Péclet number is to be interpolated such as to ensure the differences in thermal conductivity in the fluid and solid domain is modelled correctly, as well as maintaining the correct scaling of the two seperate, but similar, PDEs governing the heat transfer in the two domains. It is proposed that the Péclet number be interpolated by some function multiplied by the Péclet

number of the fluid:

$$Pe(\gamma) = f(\gamma)\, Pe_f \;\text{ for }\; 0 \leq \gamma \leq 1 \tag{4.3}$$

By simultaneously non-dimensionalising the heat conduction equation for a solid domain and the convection-diffusion equation for a fluid domain, see appendix D, it can be shown that the Péclet number, or dimensionless inverse conductivity, for the solid domain can be defined as follows:

$$Pe_s = C_k Pe_f \tag{4.4}$$

where $C_k = \frac{k_f}{k_s}$ is the ratio between the thermal conductivities of the fluid and solid domains. This implies that the interpolation function, $f(\gamma)$, needs to satisfy the bounds:

$$\text{Solid:} \qquad f(0) = C_k \tag{4.5a}$$

$$\text{Fluid:} \qquad f(1) = 1 \tag{4.5b}$$

A list of the thermal conductivity ratios for several material and fluid combinations can be found in appendix D.

Before setting up the interpolation function, it is important to consider what properties one is interpolating and how this may have to be penalised. The Péclet number is representative of the inverse of the conductivity, thus the effective conductivity can be seen as $\frac{1}{Pe}$. When considering objective functions that depend on the heat transfer in the solid domain, it is likely that the conductivity of intermediate design variables needs to be penalised. Thus, $f(\gamma)$ needs to be chosen in such a way that $f^{-1}(\gamma)$ becomes convex with a controllable penalisation factor.

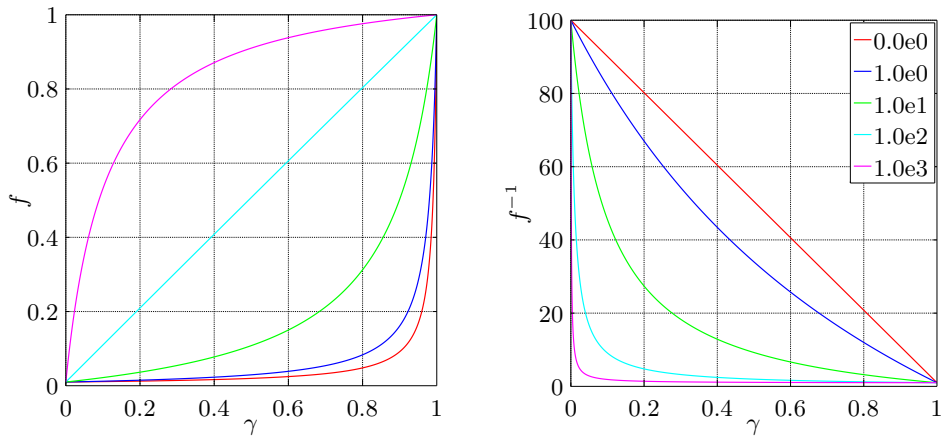
A RAMP function is chosen for the interpolation of the effective conductivity, $\frac{1}{Pe}$, and the inverse of the interpolation function in equation (4.3) is therefore defined as:

$$\begin{aligned} f^{-1}(\gamma) &= 1 + \left(\frac{1}{C_k} - 1\right) \frac{1-\gamma}{1+\gamma q_f} \\ &= \frac{\gamma(C_k(1+q_f)-1)+1}{C_k(1+q_f\gamma)} \;\text{ for }\; 0 \leq \gamma \leq 1 \end{aligned} \tag{4.6}$$

where $q_f \geq 0$ is the penalisation factor and it has been assumed that $C_k \leq 1$ with respect to correct penalisation, which is physically very reasonable as the conductivity of solid material, especially for metals, is significantly larger than for fluids. The actual interpolation function used in equation (4.3) then becomes:

$$f(\gamma) = \frac{C_k(1+q_f\gamma)}{\gamma(C_k(1+q_f)-1)+1} \;\text{ for }\; 0 \leq \gamma \leq 1 \tag{4.7}$$

which satisfies the bounds in equation (4.5).



(a) Interpolation function **(b)** Inverse function

Figure 4.3 – Plot of the Péclet interpolation function, equation (4.7), for $C_k = 10^{-2}$ and various values of the penalty factor, q_f.

Penalisation parameter	Property
$q_f = {C_k}^{-1}$	linear
$q_f < {C_k}^{-1}$	convex
$q_f > {C_k}^{-1}$	concave

(a) $f(\gamma)$

Penalisation parameter	Property
$q_f = 0$	linear
$q_f > 0$	convex

(b) $f^{-1}(\gamma)$

Table 4.1 – Properties of the interpolation function for the Péclet number and its inverse.

Table 4.1 and figure 4.3 show the properties of the interpolation function and its inverse. For good measure, the properties of the interpolation function are described even though it is not of direct interest, as it is the effective conductivity, and thus the inverse function, that is the physical property of interest.

An investigation of the interpolation penalisation factor is carried out in appendix H in the context of various objective functional candidates.

4.2.3 Interpolation of the Boussinesq forcing

To the authors knowledge, no papers exists in the literature treating the topology optimisation of natural convection problems. The only paper that comes close, is the very recent paper by Deng et al. [24] treating the to-

pology optimisation of steady and unsteady incompressible Navier-Stokes flows driven by body forces. This is to the authors knowledge the only paper treating flow problems driven by body forces and this is done through a design-dependent volumetric body force.

Deng et al. [24] state that the body forces need to be dependent on the design variables and zero inside the solid parts of the domain, because a non-zero body force inside the solid domain will force the fluid to flow through it. Initally, this was thought to be a very likely problem when topology optimising natural convection problems. However, this did not appear to be a problem when tested for a simple test problem of a differentially heated square cavity with a porous bar, with design variable dependent impermeability, seperating the cavity into two smaller rectangular cavities. For this problem, it actually appeared that decreasing the Boussinesq forcing, in proportion to the design variable, caused the Brinkman penalisation to act as a forcing in itself for low impermeabilities, causing large velocities and weird flow conditions instead of translational-symmetry as expected. It is postulated that this may have been due to the interpolation of the conductivity and the Boussinesq forcing not being correlated in respect to one another. Finally, as having a constant Boussinesq forcing term with respect to design variable did not introduce this anomalous behaviour, the Boussinesq forcing term is left as independent of the design variable for the optimisations carried out in this thesis.

However, further investigation into the behaviour of the Boussinesq forcing for variable densities should be made before clear conclusions can be drawn.

4.3 Solving the topology optimisation problem

A general constrained topology optimisation problem can be written as follows:

$$\underset{\boldsymbol{\gamma},\mathbf{s}}{\text{minimise:}} \quad f(\boldsymbol{\gamma},\mathbf{s}) \tag{4.8a}$$

$$\text{subject to:} \quad \boldsymbol{\mathscr{R}}(\boldsymbol{\gamma},\mathbf{s}) = \mathbf{0} \tag{4.8b}$$

$$h_i(\boldsymbol{\gamma}) = 0 \qquad \text{for } i = 1,..,m_{eq} \tag{4.8c}$$

$$g_i(\boldsymbol{\gamma}) \geq 0 \qquad \text{for } i = 1,..,m_{in} \tag{4.8d}$$

$$\gamma_i^l \leq \gamma_i \leq \gamma_i^u \qquad \text{for } i = 1,..,n_{\mathrm{d}} \tag{4.8e}$$

where $\boldsymbol{\gamma}$ is the vector of design variables, $\mathbf{s}$ is the vector of state variables of the discretised PDE, f is the objective function, g_i is the m_{eq} number of scalar equality constraints, h_i is the m_{in} number of scalar inequality constraints, γ_i^l and γ_i^u are the lower and upper values for the box constraints on the n_{d} number of design variables and $\boldsymbol{\mathscr{R}}$ is the vector of residuals for the

discretised PDE:

$$\mathscr{R}(\boldsymbol{\gamma}, \mathbf{s}) = \mathbf{M}(\boldsymbol{\gamma}, \mathbf{s})\,\mathbf{s} - \mathbf{b}(\boldsymbol{\gamma}, \mathbf{s}) \tag{4.9}$$

which in this thesis is given by equations (3.4) and (3.5). Notice that optimisation problem (4.8) is posed as a minimisation problem with both the design and state variables as optimisation variables. This is because no assumptions have been made on how the state problem is solved. There are two ways to solve the optimisation problem; the SAND approach and the nested approach. SAND stands for 'simultaneous analysis and design' and this approach uses both the design and state variables as optimisation variables simultaneously. The nested approach is the more traditional approach to solving topology optimisation problems, where the state variables are solved for independently and a simplified optimisation problem is posed with only the design variables as optimisation variables:

$$\begin{aligned}
\underset{\boldsymbol{\gamma}}{\text{minimise:}} \quad & f(\boldsymbol{\gamma}, \mathbf{s}) && && \text{(4.10a)}\\
\text{subject to:} \quad & \mathscr{R}(\boldsymbol{\gamma}, \mathbf{s}) = \mathbf{0} && && \text{(4.10b)}\\
& h_i(\boldsymbol{\gamma}) = 0 && \text{for } i = 1, .., m_{eq} && \text{(4.10c)}\\
& g_i(\boldsymbol{\gamma}) \geq 0 && \text{for } i = 1, .., m_{in} && \text{(4.10d)}\\
& \gamma_i^l \leq \gamma_i \leq \gamma_i^u && \text{for } i = 1, .., n_{\mathrm{d}} && \text{(4.10e)}
\end{aligned}$$

under the assumption that the state solution, $\mathbf{s}$, is found from solving the state equations, in this case through finite element analysis. The reason as to why the nested approach is most often used, is if one already has access to a pre-existing finite element modelling tool to solve for the state variables and also that the SAND approach often requires more computational time and memory, as the problem size is extended by the number of state variables. Furthermore, if the optimisation is stopped prematurely, the state variables might very well not fulfill the state equations.

In general, topology optimisation problems are nonlinear and nonconvex. The problems exhibit many local minima and thus, when solving the constrained topology optimisation problems, the converged solution is generally likely to be a local minima and strongly dependent on the initial guess. See appendix C for further information.

4.3.1 The Method of Moving Asymptotes (MMA)

The Method of Moving Asymptotes (MMA) by Svanberg [53, 54, 55] is an optimisation algorithm for general nonlinear constrained problems, that is very popular in the structural, in particular topology, optimisation community. Although the widespread use of MMA in the community mostly stems from tradition, this tradition has its foundation in sound performance. The MMA algorithm handles problems with a large number of design

variables, subject to few constraint functions, very well and this is exactly the type of problems that arise in most topology optimisation applications.

The optimisation results in this thesis are obtained using a parallelised version of the MMA algorithm, as described in [4]. Another advantage of using the MMA algorithm for large-scale topology optimisation problems is that the method uses seperable convex approximations, which makes MMA well suited for parallelisation.

4.3.2 Adjoint sensitivities

In order to apply gradient-based optimisation algorithms to topology optimisation problems, and design optimisation in general, the gradients of the objective function and any given constraint functions, with respect to the design variables, need to be known. This can either be through analytical methods, numerical methods or a combination thereof. These derivatives are often known as sensitivities.

In topology optimisation, the adjoint method is the standard tool for calculating the sensitivities of the various optimisation functionals. The adjoint method is a smart way to calculate design senstitivities when the number of design variables is larger than the number of constraint functions. This is almost exclusively the case in topology optimisation, as the number of design variables is equal to the number of finite elements in the model or the number of nodes in the model. Thus, even for smaller problems, the number of design variables quickly exceeds the number of constraint functions. It should be noted that in topology optimisation, there are of course as many box constraints as there are design variables, but these are handled easily within most optimisation algorithms.

The adjoint method can be split into two approaches; the continuous adjoint approach and the discrete adjoint approach. The two approaches differ in the sequence of the steps performed in order to model the adjoint problem. The continuous adjoint approach is also known as the "optimise-then-discretise" approach, where the state problem, the adjoint problem and all optimisation functionals are kept on a continuous level until the very end, where they are discretised after the optimality conditions have been found. The continuous approach yields an adjoint PDE, which can be discretised using any suitable discretisation scheme. The discrete adjoint approach is also known as the "discretise-then-optimise" approach, where the state problem is discretised and then the adjoint problem is derived from the discrete system of equations. Thus, the adjoint problem inherits the discretisation scheme of the original state problem. See for example the paper by Giles and Pierce [30] for a comparison of the discrete and continuous approaches.

The discrete adjoint approach is described in detail in the following section and the continuous adjoint approach is shortly described in appendix F.1.

The discrete adjoint approach

For several reasons, the discrete adjoint approach is used throughout this thesis. First of all, the discrete approach is very easy to derive on a linear algebra level; that is, working with the system matrices and vectors. Furthermore, it is very easy to implement when one has full control over and insight into the underlying finite element formulation and implementation. Lastly, the discrete adjoint approach yields "exact" sensitivities for the discrete optimisation problem [30].

Firstly, a general functional is defined as:

$$\Phi(\boldsymbol{\gamma}, \mathbf{s}(\boldsymbol{\gamma})) \tag{4.11}$$

which is a function of both the design variables, $\boldsymbol{\gamma}$, and the state variables, $\mathbf{s}$. The state variables are the solution to the discrete system of equations defined by the residual:

$$\boldsymbol{\mathscr{R}}(\boldsymbol{\gamma}, \mathbf{s}(\boldsymbol{\gamma})) = \mathbf{0} \tag{4.12}$$

Due to the state variables implicitly depending on the design variables, the total derivative is taken of the functional:

$$\underset{1\times n_\mathrm{d}}{\frac{d\Phi}{d\boldsymbol{\gamma}}} = \underset{1\times n_\mathrm{d}}{\frac{\partial\Phi}{\partial\boldsymbol{\gamma}}} + \underset{1\times n_\mathrm{s}}{\frac{d\Phi}{d\mathbf{s}}} \overset{n_\mathrm{s}\times n_\mathrm{d}}{\frac{\partial\mathbf{s}}{\partial\boldsymbol{\gamma}}} \tag{4.13}$$

where $\frac{d\square}{d\square}$ denotes the total derivative. The total derivative is defined as:

$$\frac{da}{db} = \frac{\partial a}{\partial b} + \frac{\partial a}{\partial c_i}\frac{\partial c_i}{\partial b} \tag{4.14}$$

where $\frac{\partial\square}{\partial\square}$ denotes the partial derivative and c_i denotes any variables that a explicitly depends on, which are implicit functions of b. In order for the residual to fulfill equation (4.12) after a finite change of the design variables, the total derivative of the residual with respect to the design variables is required to be zero:

$$\underset{n_\mathrm{s}\times n_\mathrm{d}}{\frac{d\boldsymbol{\mathscr{R}}}{d\boldsymbol{\gamma}}} = \underset{n_\mathrm{s}\times n_\mathrm{d}}{\frac{\partial\boldsymbol{\mathscr{R}}}{\partial\boldsymbol{\gamma}}} + \underset{n_\mathrm{s}\times n_\mathrm{s}}{\frac{d\boldsymbol{\mathscr{R}}}{d\mathbf{s}}} \underset{n_\mathrm{s}\times n_\mathrm{d}}{\frac{\partial\mathbf{s}}{\partial\boldsymbol{\gamma}}} = \mathbf{0} \tag{4.15}$$

The derivative of the state variables with respect to the design variables can then be found to be:

$$\underset{n_\mathrm{s}\times n_\mathrm{d}}{\frac{\partial\mathbf{s}}{\partial\boldsymbol{\gamma}}} = -\left(\underset{n_\mathrm{s}\times n_\mathrm{s}}{\frac{d\boldsymbol{\mathscr{R}}}{d\mathbf{s}}}\right)^{-1} \underset{n_\mathrm{s}\times n_\mathrm{d}}{\frac{\partial\boldsymbol{\mathscr{R}}}{\partial\boldsymbol{\gamma}}} \tag{4.16}$$

which can be inserted into equation (4.13) to yield:

$$\underset{1\times n_\mathrm{d}}{\frac{d\Phi}{d\boldsymbol{\gamma}}} = \underset{1\times n_\mathrm{d}}{\frac{\partial\Phi}{\partial\boldsymbol{\gamma}}} - \underset{1\times n_\mathrm{s}}{\frac{\partial\Phi}{\partial\mathbf{s}}} \underset{n_\mathrm{s}\times n_\mathrm{s}}{\left(\frac{d\boldsymbol{\mathscr{R}}}{d\mathbf{s}}\right)^{-1}} \underset{n_\mathrm{s}\times n_\mathrm{d}}{\frac{\partial\boldsymbol{\mathscr{R}}}{\partial\boldsymbol{\gamma}}} \tag{4.17}$$

By defining the so-called adjoint problem as:

$$\overset{n_\mathrm{s}\times n_\mathrm{s}}{\frac{d\boldsymbol{\mathscr{R}}}{d\mathbf{s}}^{\mathrm{T}}} \underset{n_\mathrm{s}\times 1}{\boldsymbol{\lambda}} = \overset{n_\mathrm{s}\times 1}{\frac{\partial\Phi}{\partial\mathbf{s}}^{\mathrm{T}}} \tag{4.18}$$

where $\boldsymbol{\lambda}$ is the vector of adjoint variables, the sensitivities can easily be calculated as:

$$\overset{1\times n_\mathrm{d}}{\frac{d\Phi}{d\boldsymbol{\gamma}}} = \overset{1\times n_\mathrm{d}}{\frac{\partial\Phi}{\partial\boldsymbol{\gamma}}} - \underset{1\times n_\mathrm{s}}{\boldsymbol{\lambda}^{\mathrm{T}}} \overset{n_\mathrm{s}\times n_\mathrm{d}}{\frac{\partial\boldsymbol{\mathscr{R}}}{\partial\boldsymbol{\gamma}}} \tag{4.19}$$

where the derivatives of the objective function and residual vector, with respect to the design variables, can be easily found either analytically or using finite differences. Throughout this thesis, the derivatives are derived analytically due to complete access to the underlying finite element formulation and these can be found in appendix G. If one does not have access to information about the underlying discretisation, finite differences could be used. Furthermore, an analogous derivation of the adjoint variables comparing them to Lagrange multipliers for PDE constrained optimisation problems can be found in appendix F.2.

It can be seen that the adjoint problem (4.18) depends on the transpose of the tangent system matrix of the original state problem. For many problems, this matrix is symmetric and the factorisation from solving the state problem can be reused. However, the FEM discretised flow equations result in an unsymmetric tangent system matrix and hence, the transposed matrix must be calculated and factorised before solving the adjoint problem. However this is not significant as, due to the nonlinearity of the underlying problem, the matrix would have to be recalculated and factorised anyway, in order to use the fully converged state solution in the nonlinear tangent system matrix, ensuring as accurate sensitivities as possible.

It should be noted that sensitivities of any constraint functionals, to be imposed on the optimisation problem, are found using the same methodology. A new adjoint variable field is introduced per functional and thus problem (4.18) needs to be calculated once per functional. Here the factorised transposed tangent system matrix can be reused.

4.3.3 Filtering

There are several reasons as to why some form of filtering of the design distribution is beneficial for some topology optimisation problems. Fluid

flow problems where the objective is to minimise the dissipated energy are generally well-posed and no filtering is needed [14], but for structural mechanics and heat transfer problems a problem that exists is the appearance of the so-called checkerboards in the design solution. Alternating solid and void elements create areas of solid elements connected at the corners only. These occur due to the fact that the optimisation procedure always exploits faults and weaknesses in the numerical modelling. The checkerboard pattern results in a structure that is artificially stiff or conductive, and the optimisation procedure, therefore, takes advantage of this. Another closely related problem is the appearance of single-node pivots as seen in figure 4.4.

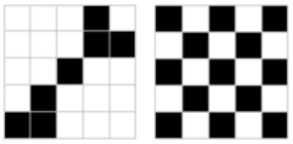


Figure 4.4 – Illustration of checkerboards and single-node pivots.

Another reason for the need of filtering, is to create mesh-independent solutions as it introduces a lengthscale into the design. Topology optimisation for structural mechanics and heat transfer results in mesh-dependent solutions as a finer mesh allows for the generation of finer truss-like solutions with more holes in them, which generally increases the efficiency of the material use. This is not desirable as an increased resolution and accuracy of the FEM solution then will not go hand-in-hand with an increased resolution and accuracy of the *same* optimal solution.

There are many different ways to fix the complications listed above, most of which are described in the book by Bendsøe and Sigmund [13] and the comprehensive investigation of various filter types in the paper by Sigmund [49]. The filtering throughout this report is done by looking at the "neighbourhood" of the individual e'th element, which is defined below as the set of elements with centres within the filter radius, R:

$$N_e = \{i \mid \|\mathbf{x}_i - \mathbf{x}_e\| \leq R\} \tag{4.20}$$

where R is the filter radius and $\mathbf{x}_i$ is the spatial location of the element i.

Density filtering

Filtering of design variables, or densities, is done by updating the element densities with a "weighted average" of the densities of the elements within the predefined neighbourhood (equation 4.20). The densities are updated

using:

$$\tilde{\gamma}_e = \frac{\sum_{i \in N_e} w(\mathbf{x}_i) v_i \gamma_i}{\sum_{i \in N_e} w(\mathbf{x}_i) v_i} \tag{4.21}$$

where v_i is the volume of element i and $w(\mathbf{x}_i)$ is a weighting function, which throughout this thesis is defined as the following linearly decaying, cone-shaped, function:

$$w(\mathbf{x}_i) = R - \|\mathbf{x}_i - \mathbf{x}_e\| \tag{4.22}$$

The filtered densities, $\tilde{\boldsymbol{\gamma}}$, are now the physically meaningful variables, whereas $\boldsymbol{\gamma}$ is now just a non-physical design variable. As it is $\boldsymbol{\gamma}$ that is updated using the optimisation algorithm, the sensitivities have to be updated using the chain rule:

$$\frac{\partial f}{\partial \gamma_e} = \sum_{i \in N_e} \frac{\partial f}{\partial \tilde{\gamma}_i} \frac{\partial \tilde{\gamma}_i}{\partial \gamma_e} \tag{4.23}$$

where the sensitivity of the filtered density with respect to the non-physical design variable is:

$$\frac{\partial \tilde{\gamma}_i}{\partial \gamma_e} = \frac{w(\mathbf{x}_e) v_e}{\sum_{j \in N_i} w(\mathbf{x}_j) v_j} \tag{4.24}$$

Although filtering solves the problems of checkerboarding and mesh-dependent solutions, it also introduces a band around the edge of the solution where the design transitions from one phase to the other. This can be solved by using projection methods and robust formulations [49, 64] in order to have crisp final designs with a clear seperation between material and void, but time has unfortunately not allowed this to be implemented.

It is important to note that for all the figures showing the design fields, it is the physically relevant filtered variables that are shown.

4.3.4 Continuation approach

A continuation approach is where the optimisation problem is progressively solved for a decreasing or increasing parameter. In topology optimisation for structural mechanics and heat transfer, it is common to use a continuation approach to progressively increase the penalisation of the stiffness and conductivity. In topology optimisation for fluid flow problems, it is common to use a continuation approach to progressively increase the impermeabilities of intermediate design variables in order to push them towards zero.

By having a low or no penalisation, the design variables are often distributed throughout the range between 0 and 1, and by increasing the penalisation, the design variables are pushed towards the extremities of this range. The idea is then to start with a well-posed and smooth problem at

the beginning, with well scaled sensitivities, and then progressively increase the penalisation pushing the design variables towards 0 and 1. However, if the optimisation is started with a large penalisation value to begin with, there is usually a large risk of getting stuck in poor local minima.

It should be noted that a continuation approach does not neccessarily guarantee ending in a pure 0-1 design or a design that is better than achieved by constant penalisation [50], but this is often the case in practice.

For all of the results presented in Part II, a continuation approach has been used to acheive the results. The continuation approach consists of five steps, where the penalisation factors for the porosity and effective conductivity is changed simultaneous after 50 iterations, or at convergence, as follows:

$$q_\alpha = \left\{10^4, 10^3, 10^2, 10^1, 10^0\right\} \tag{4.25a}$$
$$q_f = \left\{10^0, 10^1, 10^2, 10^3, 10^4\right\} \tag{4.25b}$$

When the parameters are changed, the optimisation algorithm is reset as the new parameter set represents a new optimisation problem. For the final step, the optimisation algorithm is run until convergence. For all of the results presented in this thesis, the convergence criteria for the optimisation algorithm is defined as when the maximum relative change in design variables reaches below 1 percent.

4.3.5 Thermal compliance

The thermal compliance has been succesfully used as the objective functional in the topology optimisation of pure conductive heat transfer problems, e.g. [13] and by the author in an earlier project [5]. Based on this and the investigations made in appendix H, the thermal compliance has been chosen as the objective function used throughout the results presented in this thesis.

The thermal compliance is defined as:

$$\Phi(\mathbf{t}) = \mathbf{f}_t{}^T\mathbf{t} \tag{4.26}$$

where $\mathbf{t}$ is the vector containing the nodal temperatures and $\mathbf{f}_t$ is the heat flux vector from the finite element discretised equations. By minimising the thermal compliance, the temperatures at the areas where heat flux is applied are minimised and the optimal structure will therefore be one that maximises the transport of thermal energy away from the point where the heat flux enters.

Furthermore, a constraint on the solid volume fraction is imposed on the problems, because when the conductivity of the solid material is higher than that of the fluid, $C_k < 1$, it has been observed that filling the design domain with solid material is often desirable. The solid volume fraction constraint

function is posed as follows:

$$f_{vc} = \frac{\sum_{e=1}^{n_e}(1-\gamma_e)\,V_e}{\mathrm{v}_f \sum_{e=1}^{n_e} V_e} - 1 \leq 0 \tag{4.27}$$

where γ_e is the design variable for element e, V_e is the volume of element e and v_f is the volume fraction imposed. The volume fraction is set to 0.5 or 50% throughout this thesis.

Part II

Results

Chapter 5

Forced convection

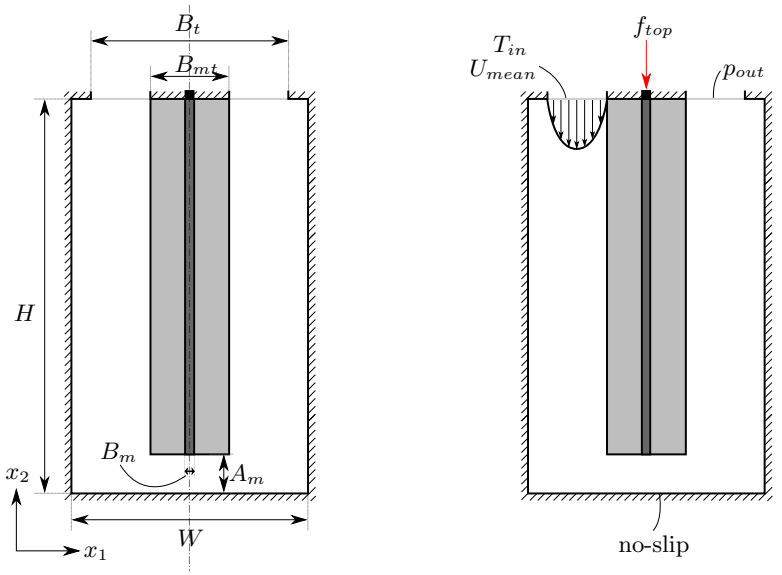
This chapter contains the optimisation results and investigations for several problems subjected to forced convection cooling.

5.1 U-bend: "Heat exchanger"

5.1.1 Problem description

The U-bend problem described here has been used to investigate the Brinkman penalisation, the interpolation functions and various candidates for objective functionals in appendix H. In this section, it is used as an optimisation problem with respect to thermal compliance. Figure 5.1 shows schematic illustrations of the layout and boundary conditions for the U-bend optimisation problem. The calculation domain consists of a U-shaped flow domain with an inlet and an outlet, seperated by a designable region in the middle. The problem is posed as a "thermal reinforcement" problem, where the optimisation goal is to optimise distribution of material around a thin solid bar of width B_m, modelled by fluid elements with the maximum Brinkman penalisation coefficient. The design domain is defined to be a B_{mt} wide area, excluding the thin solid bar in the middle. The reason for the thin solid bar being modelled by fluid elements with the maximum Brinkman penalisation coefficient, is in order to allow any problems that may occur due to fluid passing through solid regions of the design domain to show themselves.

The flow enters the domain through the left inlet with a parabolic velocity profile with a mean velocity, U_{mean}, and a given inlet temperature, T_{in}, and exits the domain through the right outlet, where a homogenous pressure distribution is prescribed, p_{out}. The rest of the domain boundary is thermally insulated, $f_n = 0$, and has no-slip conditions imposed, $u_i = 0$. A concentrated flux, f_{top}, is imposed on a small fully solid square of material, modelled by solid elements with only temperature degrees of freedom, at the top middle of the thin solid bar. Table 5.1 lists the dimensionless quantities specifying the layout and boundary conditions of the U-bend problem. All



(a) Sizes **(b)** Boundary conditions

Figure 5.1 – Schematic illustration of the sizes and boundary conditions for the U-bend problem. Black denotes fully solid, dark grey denotes elements with maximum Brinkmann penalisation coefficient and light grey denotes the design domain.

of the quantities specified are kept constant throughout. All spatial dimensions are relative to the height of the domain, H, the flow velocities are relative to the mean of the inlet flow, U_{mean}, and the flux and temperature are relative to the scales defined in appendix E. The Reynolds and Péclet numbers are thus defined by the mean inlet velocity and the height of the U-bend.

The computational domain, excluding the fully solid square, is discretised using 60×100 square elements[1], whereof the design domain and the solid bar is discretised using 20×90. The fully solid square is discretised using 2×2 elements with only temperature DOFs. The total number of velocity and pressure DOFs is 49288 and the total number of temperature DOFs is 12334.

The filter radius is set to 0.06, which is 1.2 times the element size and the maximum Brinkman penalisation coefficient is kept constant at $\overline{\alpha} = 10^6$ unless otherwise stated.

[1]It should be noted that the implementation is three-dimensional, so square elements actually means cubic elements with the u_3 DOFs constrained to zero on all nodes. This is the case for all results presented as two-dimensional.

Sizes	
Height:	$H = 1$
Width:	$W = 0.6$
Top:	$B_t = 0.5$
	$B_{mt} = 0.2$
Porous width:	$B_m = 0.02$
Throughlet height:	$A_m = 0.1$

(a) Sizes

BCs	
Inlet:	$U_{mean} = 1$
	$T_{in} = 0$
Outlet:	$p_{out} = 0$
Flux:	$f_{top} = 10^{-2}$

(b) Boundary conditions

Table 5.1 – The dimensionless quantities used for the U-bend problem shown in figure 5.1.

Varying Pe and Re-numbers

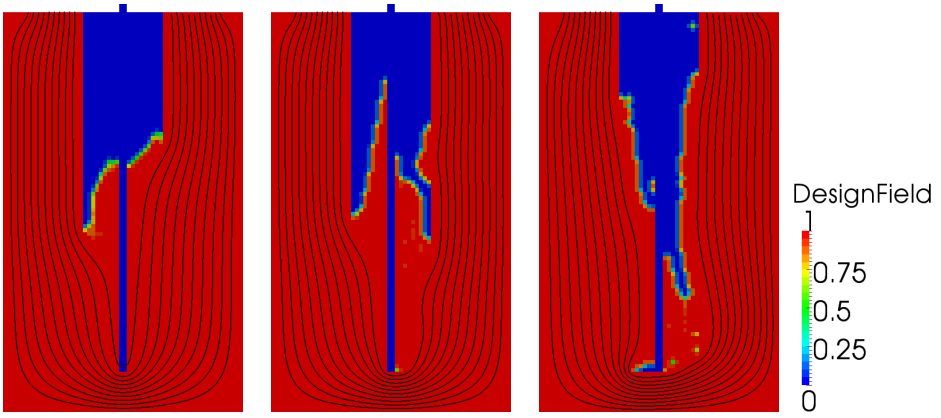
The problem is investigated for $Re = Pe = \{1, 10, 100\}$ which is equivalent to increasing the flow velocity while keeping all material properties constant, $Pr = 1$ and $C_k = 10^{-2}$.

The problem is also investigated for varying $Pe = \{1, 10, 100\}$ under constant $Re = 1$. In order for this to be equivalent to decreasing the thermal conductivity of the fluid, the thermal conductivity ratio is varied accordingly, $C_k = \left\{10^{-2}, 10^{-3}, 10^{-4}\right\}$.

It is important to note that the problem is considered as purely academic at this stage and the values of the various parameters have therefore not been chosen to represent any specific physical situation. But the values are in the range of realistic problems, so are thought to be representative of the problems that can be physically encountered.

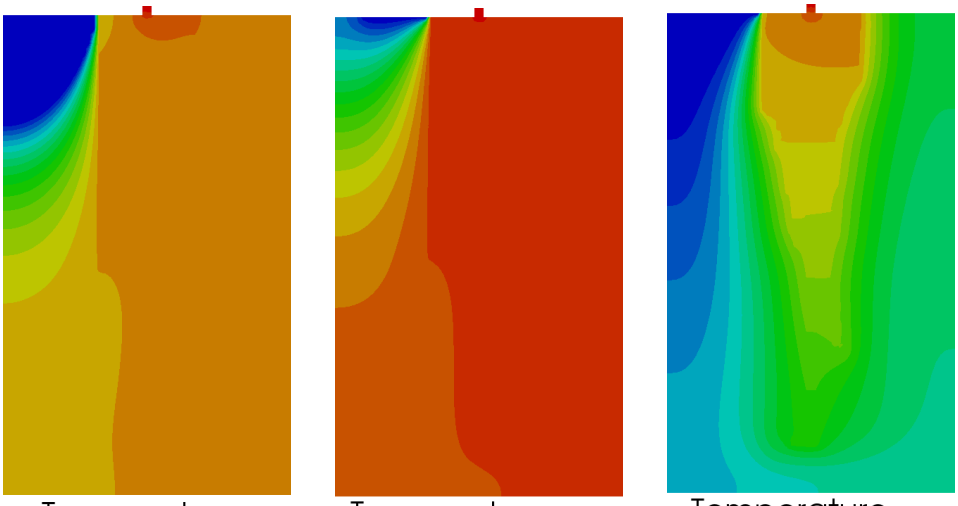
5.1.2 Varying $Re = Pe$

Figure 5.2 shows the optimised designs for the U-bend problem at varying $Re = Pe$. It can be seen that the obtained designs differ significantly from one another, which is as to be expected. The legend for the design field is here shown a single time and will not be uneccessarily repeated in this thesis to save space; blue is solid material and red is fluid. When increasing the velocity of the flow, not only does convective heat transfer become more important, but the flow also changes as the momentum of the moving fluid becomes more significant. From figure 5.2a, it can be seen that for $Re = Pe = 1$ all of the material is kept close to the top of the domain where the heat flux enters and a slightly longer wall is formed at the left-hand side where the cold fluid enters. This is likely due to the fact, that for $Pe = 1$ the thermal diffusion in the fluid is significant and the fluid does not remain cold as far into the domain as for the cases where the thermal convection is dominant, as can be seen in figure 5.3. Figure 5.2b shows that for $Re = Pe = 10$, the solid material is still kept close to the top where the



(a) $Re = Pe = 1$ **(b)** $Re = Pe = 10$ **(c)** $Re = Pe = 100$

Figure 5.2 – Optimised designs for the U-bend problem, with respect to thermal compliance, at various $Re = Pe$.
Objective function: **(a)** $f = 1.618 \cdot 10^{2}$ - **(b)** $f = 1.254 \cdot 10^{1}$ - **(c)** $f = 4.627 \cdot 10^{-1}$
Design iterations: **(a)** 140 - **(b)** 225 - **(c)** 217



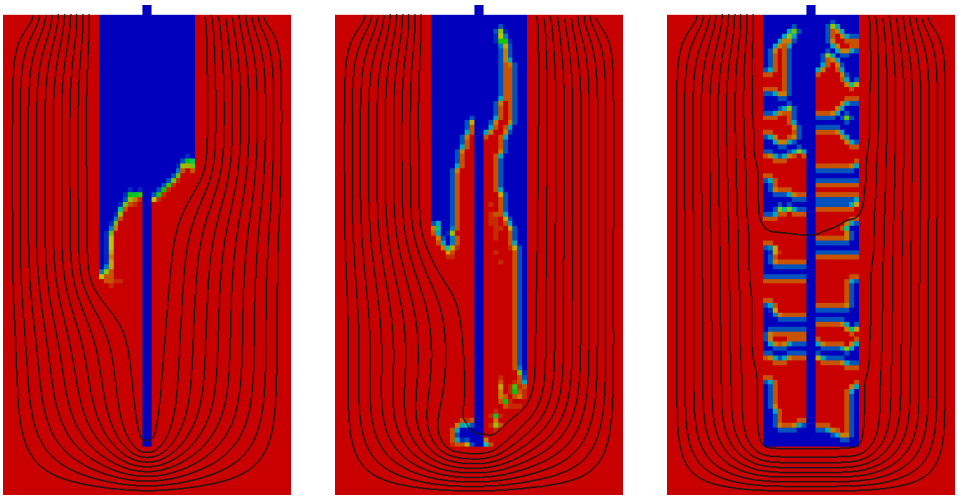
(a) $Re = Pe = 1$ **(b)** $Re = Pe = 10$ **(c)** $Re = Pe = 100$

Figure 5.3 – Temperature distribution for the optimised designs for the U-bend problem shown in figure 5.2. Note that the temperature ranges are different and the range for $Re = Pe = 1$ has been narrowed to show some detail of the distribution.

Analysis \ Optimisation	1	10	100
1	$1.618 \cdot 10^{2}$	$1.619 \cdot 10^{2}$	$1.622 \cdot 10^{2}$
10	$1.255 \cdot 10^{1}$	$1.254 \cdot 10^{1}$	$1.257 \cdot 10^{1}$
100	$4.558 \cdot 10^{-1}$	$4.506 \cdot 10^{-1}$	$4.627 \cdot 10^{-1}$

Table 5.2 – Cross-check of the optimised designs, for the U-bend problem with respect to thermal compliance, at varying $Re = Pe$, shown in figure 5.2. Coloured text highlights the minimum values for a given analysis, where blue indicates values where a given optimised design performs the best for its own analysis parameters and red indicates values where this is not the case.

heat flux enters, but the design has now spread into two branches with one on either side of the middle bar. Figure 5.2c shows that for $Re = Pe = 100$, the solid material is distributed further away from the top and spread more along the middle bar. It is also interesting to note that a small horisontal bar has been placed at the bottom-end of the middle bar, likely due to the large velocities that occur at the narrow throughlet. Table 5.2 is a cross-check table and contains the thermal compliance values for the designs shown in figure 5.2 when analysed for the analysis parameters of the other designs. The cross-checks are performed in order to shed light upon what significance one can place on the features of the designs obtained through topology optimisation. A design that has been optimised for a certain situation should perform better for its situation, than a design that has been optimised for a different situation. It can be seen from table 5.2, that the design obtained from the optimisation for $Re = Pe = 100$ actually performs worse than both of the other designs when analysed at $Re = Pe = 100$. This is likely due to the fact that the design obtained is a particularly poor local minimum and the features exhibited by the design can therefore not be attributed much significance. As can be seen from figure 5.3 and table 5.2, it is important to note that the temperatures, and thus the thermal compliance, decreases as $Re = Pe$ is increased. This is exactly as expected, as increasing $Re = Pe$ in this case is equivalent to increasing the velocity of the flow and thus increasing the convective cooling of the warm metal device.

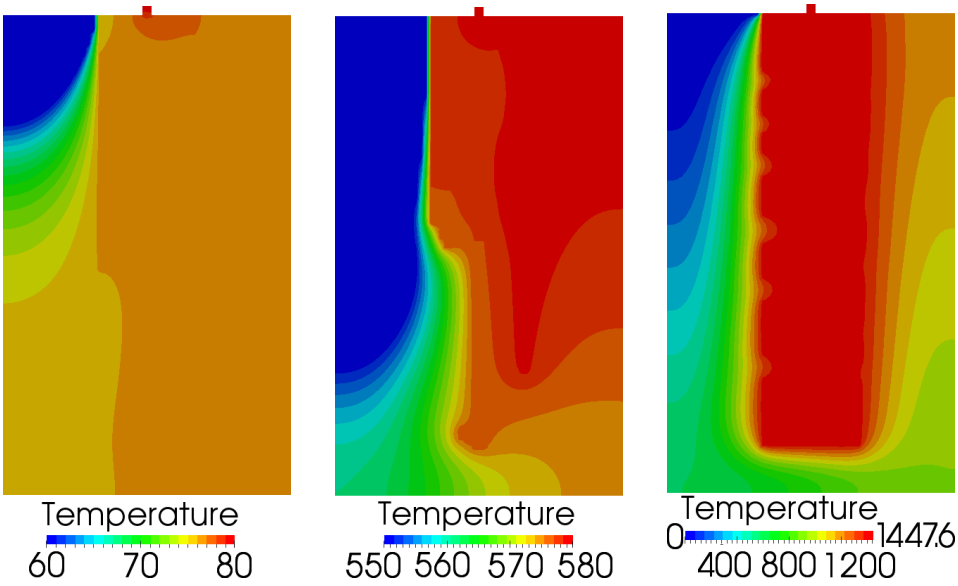


(a) $Pe = 1$ **(b)** $Pe = 10$ **(c)** $Pe = 100$

Figure 5.4 – Optimised designs for the U-bend problem, with respect to thermal compliance, at various Pe for constant $Re = 1$.
Objective function: **(a)** $f = 1.618 \cdot 10^2$ - **(b)** $f = 1.166 \cdot 10^3$ - **(c)** $f = 2.895 \cdot 10^3$
Design iterations: **(a)** 140 - **(b)** 244 - **(c)** 278

5.1.3 Varying Pe at constant $Re = 1$

Figure 5.4 shows the optimised designs for the U-bend problem at varying Pe and constant $Re = 1$. Figure 5.4a is a repetition of figure 5.2a, but is repeated here to aid a comparison of the designs. It can be seen that the obtained designs are significantly different. Figure 5.4b shows that as Pe is increased to 10, a long bar is placed at the right-hand side of the design domain, while keeping the left-hand side almost the same as for $Pe = 1$. Figure 5.4c shows that as Pe is further increased to 100 and thermal convection becomes significantly dominant, the design changes considerably. It can be seen that small "fingers", or bars, are placed on both sides along the fixed middle bar which spreads the heat into the fluid flow. As the thermal diffusivity of the fluid is now very insignificant compared to the thermal convection, the material needs to be distributed in such a way as to maximise the surface area that is in contact with the fluid flow. This is obtained by distributing the fingers along the entire middle bar in order to conduct the heat out to the flow and making their ends flat in order to increase the surface area. However, it should be noted that the magnitude of the horisontal velocity in the middle bar is approximately twice as large as for the $Pe = 1$ case and this may cause an increase in the performance of the design. This supports the claims made in appendix H, that imposing a larger minimum lengthscale in the designs may be needed in order to assure that the Brinkman penalisation is effective and that thin members, where



(a) $Re = Pe = 1$ **(b)** $Re = Pe = 10$ **(c)** $Re = Pe = 100$

Figure 5.5 – Temperature distribution for the optimised designs for the U-bend problem shown in figure 5.4. Note that the temperature ranges are different and the ranges for $Pe = \{1, 10\}$ have been narrowed to show some detail of the distribution.

the flow can pass through, are not formed.

Figure 5.5 shows the temperature distribution for the optimised design for varying Pe. It can easily be seen that as Pe is increased, thermal convection begins to dominate.

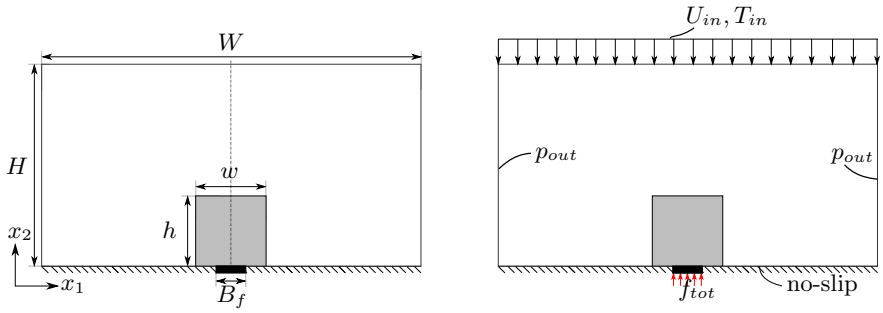
Table 5.3 shows the values for the thermal compliance obtained by a cross-check of the optimised designs for varying Pe at constant $Re = 1$. It can be seen that the designs perform exactly as they should, but it should be noted that the design for $Pe = 10$ only performs slightly better than the $Pe = 1$ design at $Pe = 10$.

		Optimisation		
		1	10	100
Analysis	1	$1.618 \cdot 10^2$	$1.622 \cdot 10^2$	$1.661 \cdot 10^2$
	10	$1.167 \cdot 10^3$	$1.166 \cdot 10^3$	$1.171 \cdot 10^3$
	100	$3.163 \cdot 10^3$	$3.066 \cdot 10^3$	$2.895 \cdot 10^3$

Table 5.3 – Cross-check of the optimised designs for the U-bend problem with respect to thermal compliance, at varying Pe, shown in figure 5.4. Blue indicates values where a given optimised design performs the best for its own analysis parameters.

Note

If the entire calculation domain is included as the design domain, solid material is simply placed right in front of the inlet. Thus, if the design domain is not restricted and kept seperate from the low temperature boundary condition, the optimisation will simply make a conductive path that connects the heat flux to the boundary condition. An alternative to restricting the design domain is to impose constraints, e.g. on the dissipated energy of the flow, that ensure a physical design.



(a) Sizes **(b)** Boundary conditions

Figure 5.6 – Schematic illustration of the sizes and boundary conditions for the heat sink problem subjected to a flow from above. Black denotes fully solid and light grey denotes the design domain.

Sizes	
Total height:	$H = 3$
Total width:	$W = 5$
Design height:	$h = 1$
Design width:	$w = 1$
Flux width:	$B_f = 1$

(a) Sizes

BCs	
Inlet:	$U_{in} = 1$
	$T_{in} = 0$
Outlet:	$p_{out} = 0$
Flux:	$f_{tot} = 2.2 \cdot 10^{-3}$

(b) Boundary conditions

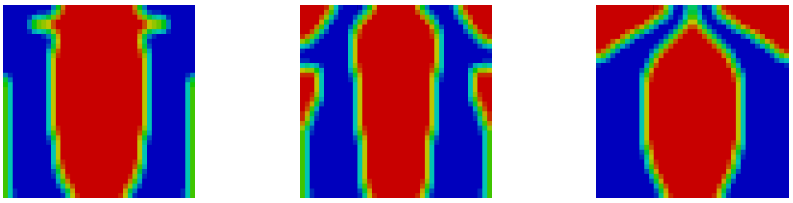
Table 5.4 – The dimensionless quantities used for the heat sink problem shown in figure 5.6.

5.2 Two-dimensional heat sink - flow from above

5.2.1 Problem description

The second problem of this section is that of a heat sink subjected to a cool flow from above. Figure 5.6 shows schematic illustations of the layout and boundary conditions for the problem. The calculation domain consists of a square design domain on top of a block of solid material that is subjected to a distributed heat flux, f_{tot}, along the bottom and a rectangular flow domain surrounding the heat sink. The flow enters the domain vertically downwards at the top edge of the domain with a constant velocity, U_{in}, and a given inlet temperature, T_{in}, and exits the domain through the left- and right-hand sides, where a homogenous pressure distribution is prescribed, p_{out}. The rest of the domain boundary is thermally insulated, $f_n = 0$, and has no-slip conditions imposed, $u_i = 0$.

Table 5.4 lists the dimensionless quantities specifying the layout and boundary conditions of the heat sink problem. All of the quantities specified are kept constant throughout, unless stated in the text. All spatial



(a) $Re = Pe = 1$ **(b)** $Re = Pe = 10$ **(c)** $Re = Pe = 100$

Figure 5.7 – Optimised designs for the heat sink problem subjected to a vertical cool flow from above at various $Re = Pe$.
Objective function: **(a)** $f = 2.264$ - **(b)** $f = 1.241 \cdot 10^{-1}$ - **(c)** $f = 6.659 \cdot 10^{-3}$
Design iterations: **(a)** 233 - **(b)** 172 - **(c)** 142

dimensions are relative to the height of the initial design domain, h, the flow velocities are relative to the inlet flow, U_{in}, and the flux and temperature are relative to the scales defined in appendix E. The Reynolds and Péclet numbers are thus defined by the inlet velocity and the initial height of the design domain.

The problem is investigated for varying $Re = Pe$ under constant parameters, $C_k = 10^{-2}$ and $\overline{\alpha} = 10^6$, and subsequently for varying parameters under constant $Re = Pe$.

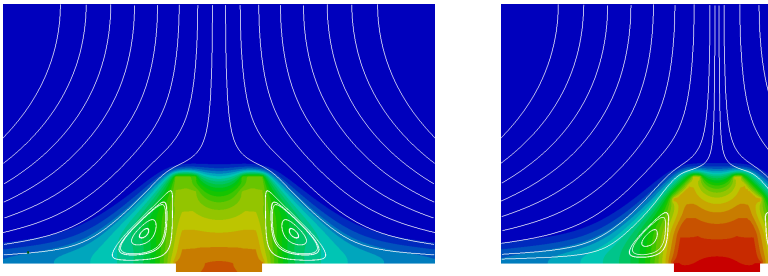
The computational domain, excluding the solid flux base, is discretised using 200×120 square elements, where the design domain is discretised using 40×40. The solid flux base is discretised using 40×4 elements with only temperature DOFs. The total number of velocity and pressure DOFs is 194568 and the total number of temperature DOFs is 48970. The filter radius is set to 0.06, which is 2.4 times the element size.

5.2.2 Varying $Re = Pe$

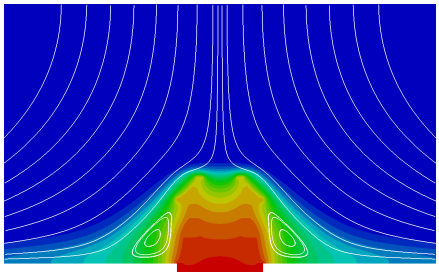
As for the U-bend problem in section 5.1, varying $Re = Pe$ while keeping other parameters constant is equivalent to increasing the flow velocity.

Figure 5.7 shows the optimised designs for a heat sink subjected to a vertical cool flow from above at various $Re = Pe$. Generally the topologies do not differ significantly; all three designs consist of a thick bar in each side of the design domain protruding to the top. However, it appears that as $Re = Pe$ is increased, the top of the bars begin to slant at $Re = Pe = 10$ and finally merge at $Re = Pe = 100$.

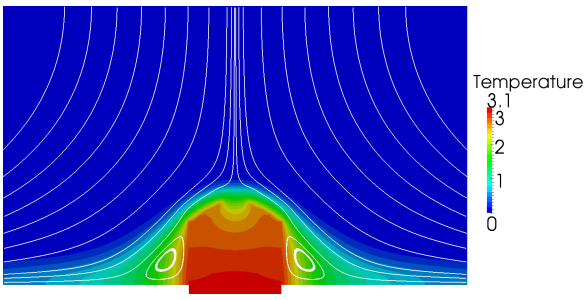
This may seem intuitive as it can be argued that as the flow velocity increases, it becomes beneficial to increase the surface area along the impinging flow that is deflected by the design domain. However, it is important and interesting to note that the design obtained from the optimisation at $Re = Pe = 1$ performs the best for all $Re = Pe = \{1, 10, 100\}$. Figure 5.8 shows the temperature fields and streamlines for the optimised designs



(a) $Re = Pe = 1$



(b) $Re = Pe = 10$



(c) $Re = Pe = 100$

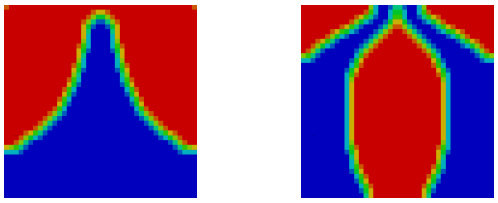
Figure 5.8 – Temperature fields and streamlines for the optimised designs shown in figure 5.7 evaluated at $Re = Pe = 100$. Objective function at $Re = Pe = 100$: **(a)** $f = 5.794 \cdot 10^{-3}$ - **(b)** $f = 6.581 \cdot 10^{-3}$ - **(c)** $f = 6.659 \cdot 10^{-3}$

shown in figure 5.7 evaluated at $Re = Pe = 100$. It can clearly be seen that the temperature of the heat sink design for $Re = Pe = 1$, figure 5.8a, is significantly lower than that of the heat sink design for $Re = Pe = 100$, figure 5.8c. This is also reflected in comparing the thermal compliance values for the three designs as stated in the figure text, where it can be seen that the designs actually progressively perform worse as the bars become slanted for increasing $Re = Pe$. It is very likely that the reason for not obtaining optimal designs for the three cases, is due to the optimisation converging to poor local minima due to the underlying optimisation problem being extremely nonlinear. Furthermore, it is postulated that the reason for the inconsistent results is due to the fact that the flow is not disturbed, and thereby changed, significantly by varying the design. This combined with the fact that distributing conductive solid material is the most effective means of quickly moving the heat away from the applied heat flux, could be the reason as for why the designs are so similar. The reason for the differences could be due to different starting points, where the amount of flow passing through the porous starting guess is different as Re is increased. Therefore, the sensitivity of the optimised designs on various parameters is studied.

Firstly, the optimisation approach was investigated. It was checked whether the aggressive continuation approach described in section 4.3.4 caused the optimisation to converge to a poor local minimum. This was done by letting the optimisation converge at each continuation step before changing the penalty parameters. However, this led to no significant differences in the optimised topologies. This, along with fact that the aggressive continuation approach leads to the optimisation converging in approximately 200-300 design iterations as compared to around 900 and above for the full continuation approach, legitimises the use of the aggressive approach.

It should also be noted that, by starting from a pure fluid or a randomly perturbed initial guess, instead of a homogenously porous one, only leads to slightly different designs.

As changes to the optimisation approach does not appear to affect the optimised topologies significantly for this problem, the effect of changing the maximum Brinkman penalisation coefficient, $\overline{\alpha} = \left\{10^2, 10^3, 10^4, 10^5, 10^6\right\}$, was investigated. For $Re = Pe = 1$ and $Re = Pe = 10$, the optimised designs were only affected very slightly, but the overall topologies remains the same as those shown in figures 5.7a and 5.7b and are therefore not shown. However for $Re = Pe = 100$, changing the maximum Brinkman penalisation coefficient from $\overline{\alpha} = 10^6$ affects the optimised designs significantly. Most of the designs are similar and consist of a triangular shape with curved sides. Figure 5.9 shows the obtained design for $\overline{\alpha} = 10^5$ in comparison to that for $\overline{\alpha} = 10^6$. It can be seen that the obtained topologies are significantly different, but their performance with respect to thermal compliance are very similar and only differ less than a percent.



(a) $\overline{\alpha} = 10^5$ **(b)** $\overline{\alpha} = 10^6$

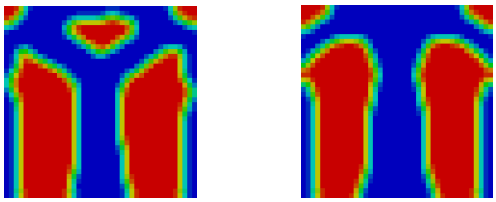
Figure 5.9 – Optimised designs for the heat sink problem subject to a vertical cool flow from above, for different maximum Brinkman penalisation coefficient, $\overline{\alpha}$.
Objective function at $\overline{\alpha} = 10^5$: **(a)** $f = 6.589 \cdot 10^{-3}$ - **(b)** $f = 6.619 \cdot 10^{-3}$
Objective function at $\overline{\alpha} = 10^6$: **(a)** $f = 6.705 \cdot 10^{-3}$ - **(b)** $f = 6.659 \cdot 10^{-3}$
Design iterations: **(a)** 196 - **(b)** 142

Following the investigations described above, it is concluded that the current problem is rather insensitive to the investigated parameters and that it seems rather "random" whether the optimisation converges to one design or the other. However, it can be seen from the fact that the design for $Re = Pe = 1$, figure 5.8a, performs best across the board, that the optimal topology for the investigated ranges seems to be determined by the conductive limit.

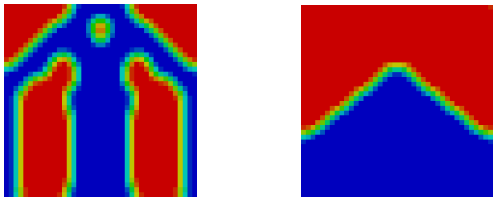
5.2.3 Varying C_k

In order to investigate the effect of the conductivity of the solid material on the optimised design, the thermal conductivity ratio is varied, $C_k = \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$, under constant $Re = Pe = 100$ and $\overline{\alpha} = 10^6$. The heat flux, f_{tot}, is varied according to equation (E.5) in appendix E, in order for the sweep to be equivalent to increasing the thermal conductivity of the solid material; $f_{tot} = 2.2 \cdot \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. The optimised design for $C_k = 10^{-2}$ has already been presented in figure 5.7c and the special case of $C_k = 1$ will be treated seperately at the end of this section.

Figure 5.10 shows the optimised designs for the heat sink problem for varying thermal conductivity ratios, $C_k = \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-1}\}$. The designs for $C_k = \{10^{-5}, 10^{-4}, 10^{-3}\}$ are very similar and share almost the same topologies, whereas the design for $C_k = 10^{-1}$ is different. It seems that for high relative conductivity of the solid material, $C_k = \{10^{-5}, 10^{-4}, 10^{-3}\}$, the optimal topology seems to be a Y-shaped member branching out to a solid shell. As the conductivity of the solid material approaches that of the fluid, $C_k = 10^{-1}$, it seems that the optimal design approaches a triangular design. As a general comment on the optimised designs presented in this section, specifically figures 5.7 and 5.10, it would be interesting to expand the filter domain to include the surrounding fluid. This would clarify whether placing material around the edges of the design domain is truly optimal or



(a) $C_k = 10^{-5}$ **(b)** $C_k = 10^{-4}$



(c) $C_k = 10^{-3}$ **(d)** $C_k = 10^{-1}$

Figure 5.10 – Optimised designs for the heat sink problem subject to a vertical cool flow from above, for different thermal conductivity ratios, C_k.
Objective function: **(a)** $f = 4.292 \cdot 10^{-3}$ - **(b)** $f = 4.559 \cdot 10^{-3}$
(c) $f = 5.248 \cdot 10^{-3}$ - **(d)** $f = 9.027 \cdot 10^{-3}$
Design iterations: **(a)** 224 - **(b)** 239 - **(c)** 300 (not fully converged) - **(d)** 216

whether it is because it is more economic for the optimisation algorithm to place material here with respect to the filter and penalisation.

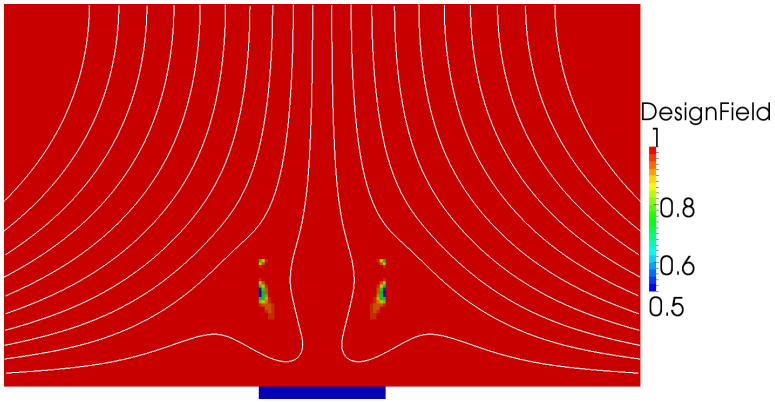


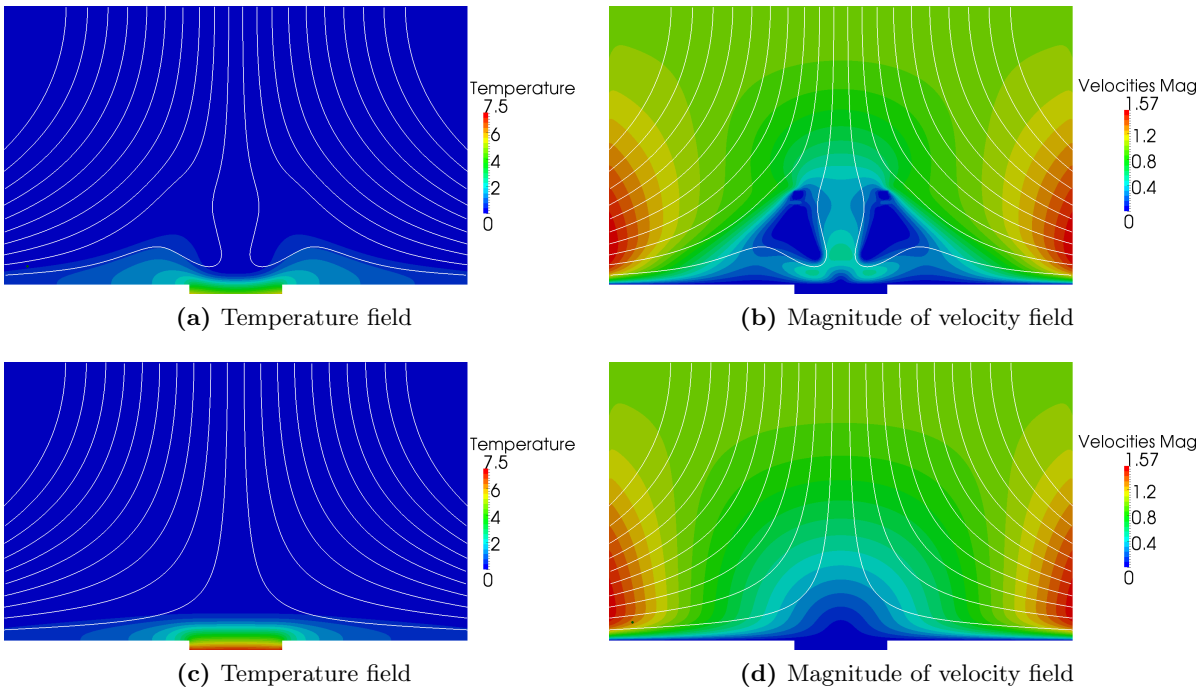
Figure 5.11 – Optimised design for the heat sink problem subjected to a vertical cool flow from above, for a solid and fluid with the same thermal conductivity, $C_k = 1$. Note that the range for the design field has been narrowed, as no elements were fully solid.
Objective function: $f = 1.107 \cdot 10^{-2}$ - Design iterations: 150 (stopped)

$C_k = 1$: Flow control problem

The special case of $C_k = 1$ is treated seperately here, due to the fact that this case if fundamentally different than the previous cases. When $C_k = 1$, the thermal conductivities of the solid and fluid are the same and there is thus no longer an advantage in placing solid material in order to conduct the heat away from the source. The problem then becomes a pure fluid flow control problem, which can be seen from the optimised design shown in figure 5.11. It can be seen that the optimised design does not utilise the maximum allowable solid volume fraction of 50%. In fact, the design only uses a solid volume fraction of 1.6% and simply places a small amount of porous material in the upper corners of the design domain in order to manipulate the flow. It is important to note that no elements are fully solid, but instead only impermeable enough to obstruct the flow in an advantageous manor. Furthermore, it should be noted that the design shown is for design iteration 150 and not the final converged design, as the optimisation began to diverge to a noisy design hereafter.

Figure 5.12 shows the temperature field and the magnitude of the velocity field for the design shown in figure 5.11 as well as that of a fully fluid domain. It can be seen that the manipulated flow field is advantageous in comparison to the undisturbed flow field and thus achieves a lower thermal compliance. It is interesting to note that the flow field in figure 5.12b shows that the optimised design collects the flow from above into an impinging jet, yielding higher velocities at the solid brick and thereby lower temperatures.

As the solid volume fraction constraint is no longer active for $C_k = 1$, it was thought to be interesting to see how the design would look for an optimisation run with a constraint on the fluid volume fraction instead.



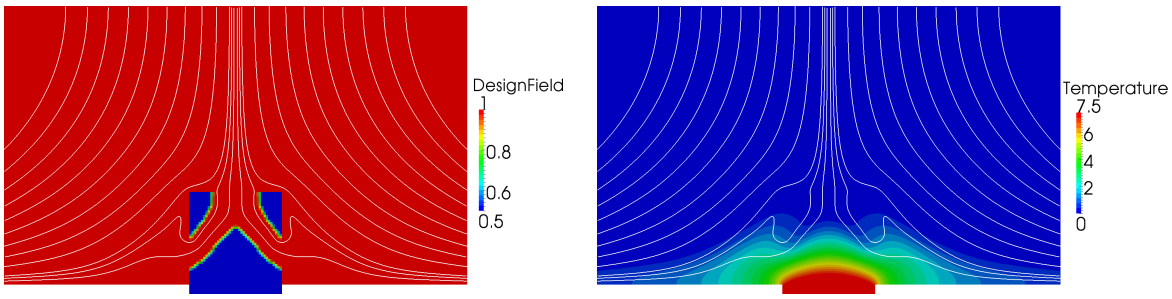
(a) Temperature field

(b) Magnitude of velocity field

(c) Temperature field

(d) Magnitude of velocity field

Figure 5.12 – Temperature and velocity fields for the heat sink problem subject to a vertical cool flow from above. **(a-b)** are for the optimised design for a solid and fluid with the same thermal conductivity, $C_k = 1$, shown in figure 5.11 and **(c-d)** are for a domain with fluid only.
Objective function: **(a)** $f = 1.107 \cdot 10^{-2}$ - **(c)** $f = 1.546 \cdot 10^{-2}$



(a) Design field

(b) Temperature field

Figure 5.13 – Optimised design and corresponding temperature field for the heat sink problem subject to a vertical cool flow from above and a constraint on the fluid volume in the design domain.
Objective function: $f = 2.416 \cdot 10^{-2}$ - Design iterations: 111

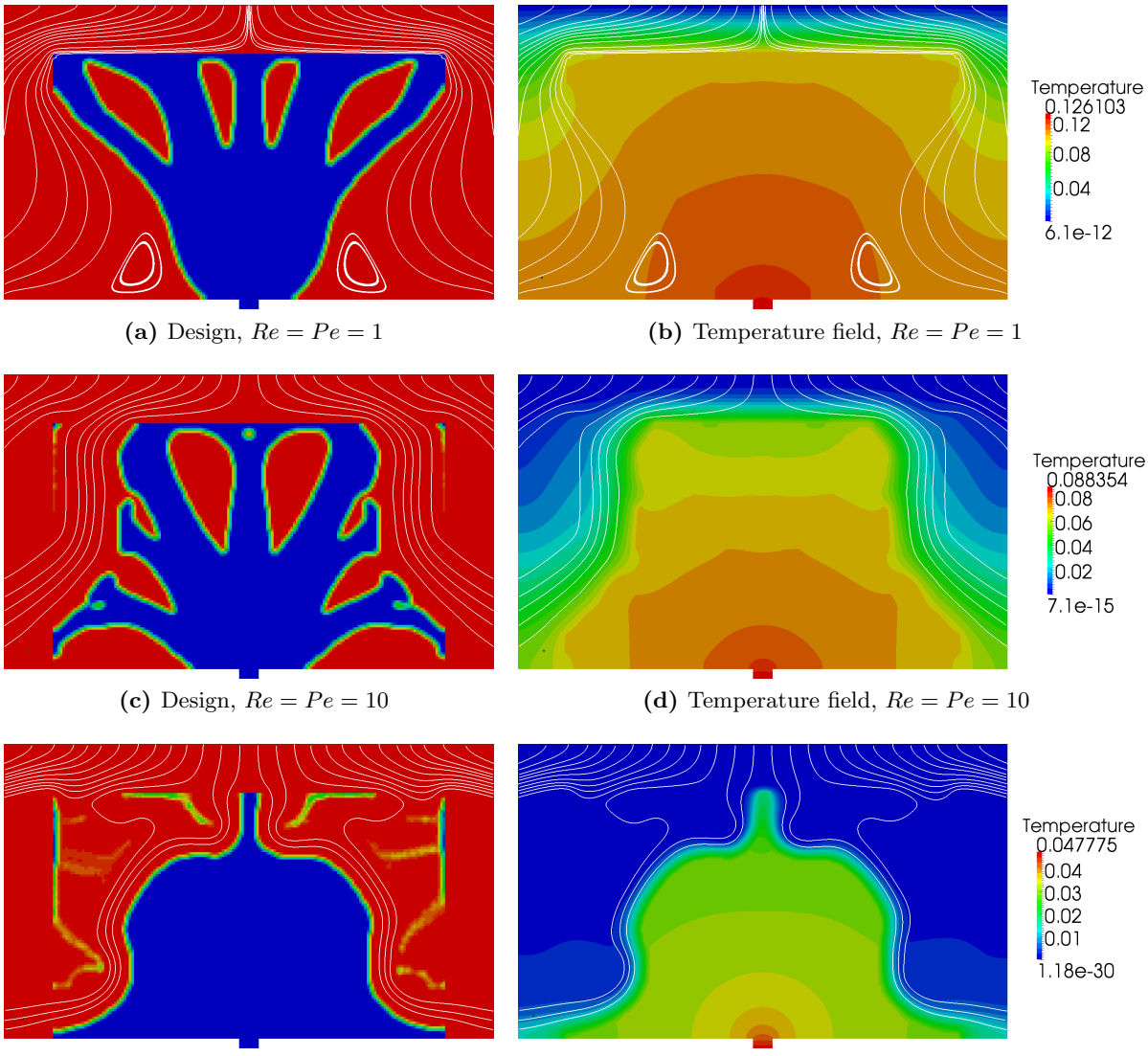
Figure 5.13 shows the optimised design and the design can be seen to collect the flow and force it onto a triangular conductive domain. It is interesting to note that the objective function for this design is larger than that of the design shown in figure 5.11. This indicates that when the solid material does not have a conductive advantage compared to the fluid, the problem becomes one of fluid control and here it is advantageous not to use too much solid material.

The physical relevance of the designs obtained for $C_k = 1$ can be questioned, as the thermal conductivity of a solid material used for cooling of a hot source almost always will be higher than that of the fluid. But the results shown here highlight the need for attributing the solid material with a different thermal conductivity than that of the fluid for thermal cooling problems where this is important.

5.2.4 Enlarged design domain

As stated in section 5.2.2, it is postulated that the reason for achieving non-optimal designs, with respect to designs for other flow conditions, is that the design domain is too small to significantly affect the overall behaviour of the surrounding flow. Therefore, the heat sink problem is investigated with a larger design domain. The solid brick at the bottom is narrowed to $B_f = 0.2$ and the design domain is enlarged to $w \times h = 4 \times 2.5$. The total number of elements and state DOFs for the computational domain, excluding the flux base, remains the same but the design domain is now discretised using 160×100. The solid flux base is discretised using 8×4 elements with only temperature DOFs.

Figure 5.14 shows the optimised designs and their corresponding temperature and flow fields. It can be seen that both designs and the flow conditions are significantly different than for the designs shown in figure 5.8. Figure 5.14 shows that as $Re = Pe$ is increased, the design contracts from a "conductive tree" for $Re = Pe = 1$ to a surface cooled dome for $Re = Pe = 100$. For $Re = Pe = 1$, where thermal diffusion is still significant, it can be seen in figure 5.14a that the design consists of a large collection of solid material at the bottom near the solid brick with members branchings out to a large flat surface at the top of the design domain. It makes sense to have this large connected surface at the top, as this is as close to the low temperature boundary as possible, thus ensuring a large surface in contact with the cool fluid. As thermal diffusion becomes less important, $Re = Pe = 10$, it can be seen in figure 5.14b that the brances are contracted to follow the flow. When thermal convection becomes dominant, $Re = Pe = 100$, it can be seen in figure 5.14c that it is no longer beneficial to have branches and the design consists of a dome, the curvature of which has adapted to the flow. It can also be seen that islands of porous material have formed that help collect and guide the flow. These are also seen in the design for $Re = Pe = 10$, but



(a) Design, $Re = Pe = 1$ **(b)** Temperature field, $Re = Pe = 1$

(c) Design, $Re = Pe = 10$ **(d)** Temperature field, $Re = Pe = 10$

(e) Design, $Re = Pe = 100$ **(f)** Temperature field, $Re = Pe = 100$

Figure 5.14 – Optimised design and corresponding temperature field for the heat sink problem subject to a vertical cool flow from above with an enlarged design domain and a narrow heat sink. Note that the temperature ranges are different. Objective function: **(a)** $f = 1.254 \cdot 10^{-3}$ - **(c)** $f = 8.764 \cdot 10^{-5}$ - **(e)** $f = 4.706 \cdot 10^{-6}$ Design iterations: **(a)** 189 - **(c)** 295 - **(e)** 330

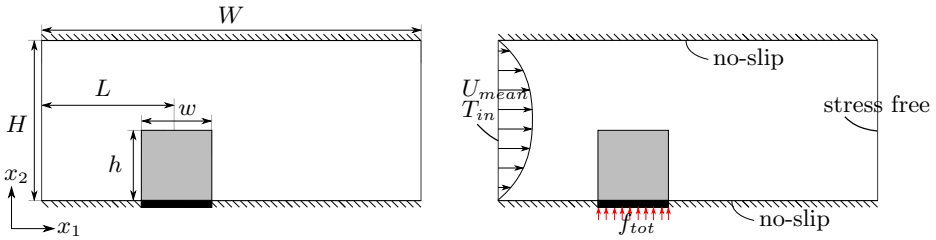
Analysis \ Optimisation	1	10	100
1	$1.254 \cdot 10^{-3}$	$1.503 \cdot 10^{-3}$	$2.535 \cdot 10^{-3}$
10	$9.104 \cdot 10^{-5}$	$8.764 \cdot 10^{-5}$	$1.770 \cdot 10^{-4}$
100	$5.269 \cdot 10^{-6}$	$4.922 \cdot 10^{-6}$	$4.706 \cdot 10^{-6}$

Table 5.5 – Cross-check of the optimised designs, for the heat sink problem subject to a vertical cool flow from above with an enlarged design domain and a narrow heat sink, shown in figure 5.14. Blue indicates values where a given optimised design performs the best for its own analysis parameters.

not as significant.

The formation of these free islands of solid material is a problem and it is likely that these could also occur for three-dimensional problems, where physically one would not be able to have such a design. Based on the flow fields observed in figure 5.14c, it is likely that these islands are not just artifacts from the optimisation process but have formed in order to guide the flow. Possible ways to remove these islands could be to have a constraint on the dissipated energy of the flow, a constraint on the conductive performance of the design or maybe a constraint on the vibration eigenvalue of the design, where the loose-hanging islands of material would lead to a low eigenvalue.

Table 5.5 shows the cross-check values for the heat sink problem with an enlarged design domain and it can be seen that the designs perform exactly as they should. This further supports the claim that the inconsistent results in section 5.2.2 could be due to the design domain being too small to significantly affect the overall behaviour of the surrounding flow.



(a) Sizes **(b)** Boundary conditions

Figure 5.15 – Schematic illustration of the sizes and boundary conditions for the heat sink problem subjected to channel flow. Black denotes fully solid and light grey denotes the design domain.

Sizes	
Total height:	$H = 3$
Total width:	$W = 8$
Design height:	$h = 1$
Design width:	$w = 1$
Design position:	$L = 2.5$

(a) Sizes

BCs	
Inlet:	$U_{mean} = 1$
	$T_{in} = 0$
Outlet:	$\boldsymbol{\sigma} \cdot \mathbf{n} = 0$
Flux:	$f_{tot} = 8.2 \cdot 10^{-3}$

(b) Boundary conditions

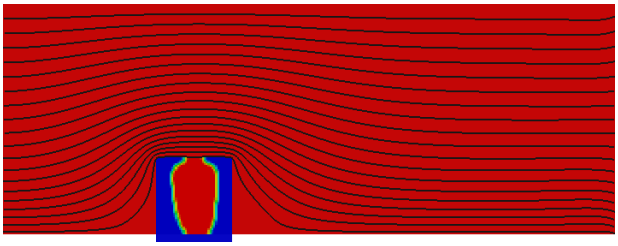
Table 5.6 – The dimensionless quantities used for the heat sink problem subjected to channel flow, shown in figure 5.15.

5.3 Two-dimensional heat sink - channel flow

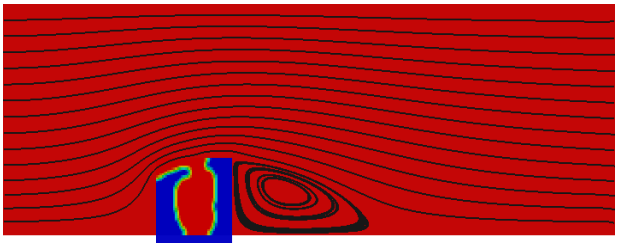
5.3.1 Problem description

The third problem of this section is a heat sink in a channel flow. Figure 5.15 shows schematic illustations of the layout and boundary conditions for the problem. The calculation domain consists of a square design domain on top of a block of solid material, subjected to a distributed heat flux, f_{tot}, along the bottom, placed inside a rectangular channel domain. The flow enters the channel at the left-hand side with a parabolic velocity profile with a mean velocity, U_{mean}, and a given inlet temperature, T_{in}, and exits the domain through the right outlet, where a stress-free outlet condition is imposed, $\boldsymbol{\sigma} \cdot \mathbf{n} = 0$. The rest of the domain boundary is thermally insulated, $f_n = 0$, and has no-slip conditions imposed, $u_i = 0$.

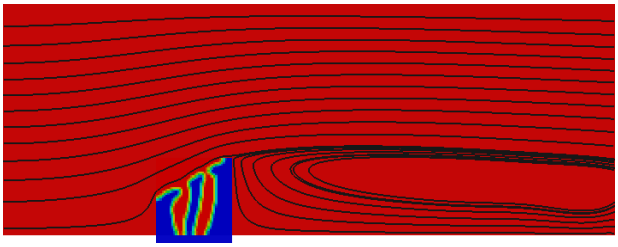
Table **??** lists the dimensionless quantities specifying the layout and boundary conditions of the heat sink problem. All of the quantities specified are kept constant throughout. All spatial dimensions are relative to the height of the design domain, h, the flow velocities are relative to the mean of the inlet flow, U_{mean}, and the flux and temperature are relative to the scales defined in appendix E. The Reynolds and Péclet numbers are thus



(a) $Re = Pe = 1$



(b) $Re = Pe = 10$



(c) $Re = Pe = 100$

Figure 5.16 – Optimised designs for the heat sink problem in a channel flow from the left, at various $Re = Pe$.
Objective function: **(a)** $f = 7.840 \cdot 10^2$ - **(b)** $f = 3.834 \cdot 10^1$ - **(c)** $f = 1.724 \cdot 10^0$
Design iterations: **(a)** 140 - **(b)** 194 - **(c)** 180

defined by the mean inlet velocity and the height of the design domain.

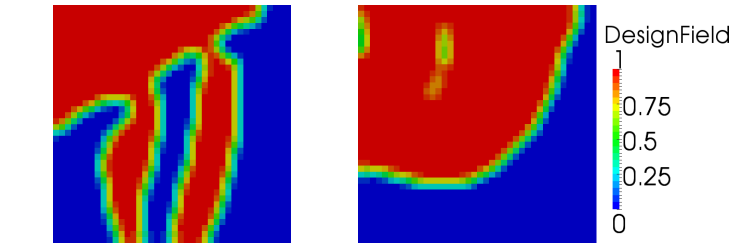
The problem is investigated for varying $Re = Pe$ under constant parameters, $C_k = 10^{-2}$ and $\overline{\alpha} = 10^6$.

The computational domain, excluding the solid flux base, is discretised using 200×120 square elements, where the design domain is discretised using 40×40. The solid flux base is discretised using 40×4 elements with only temperature DOFs. The total number of velocity and pressure DOFs is 194568 and the total number of temperature DOFs is 48970. The filter radius is set to 0.06, which is 2.4 times the element size.

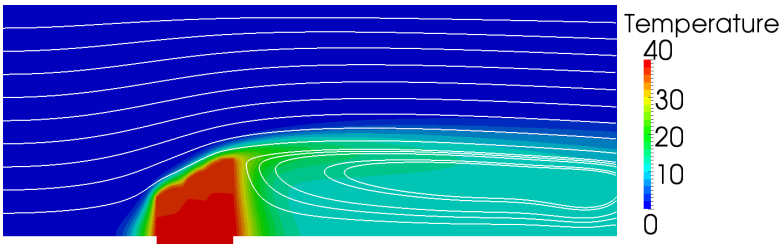
5.3.2 Varying $Re = Pe$

Figure 5.16 shows the optimised designs as well as their respective streamlines for the heat sink problem in a channel flow. It can easily be seen that, as expected for a flow over a bluff body, when Re is increased the flow seperates at the downstream corner, a recirculation zone develops and this grows in size as Re is further increased. It is interesting to note that a trend appears in the designs as $Re = Pe$ is increased. It can be seen that the top of the heat sink begins to slant to let the flow pass over the heat sink more easily. It is also important to note that for $Re = Pe = 1$, the design resembles the designs obtained for the heat sink problem subjected to a flow from above for $Re = Pe = 1$, figure 5.7a.

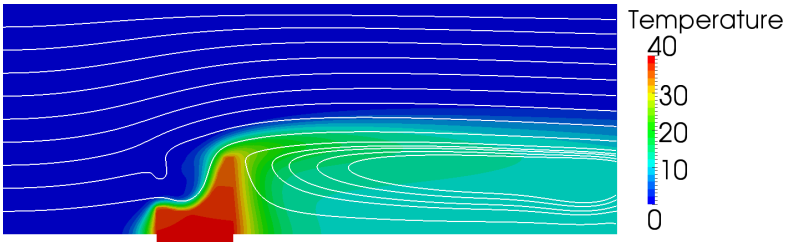
As for the flow from above, a cross-check reveals that the $Re = Pe = 1$ design actually performs the best for both $Re = Pe = 1$ and $Re = Pe = 10$. Furthermore, the $Re = Pe = 10$ design performs better than the $Re = Pe = 100$ at $Re = Pe = 100$, which means that the observed trend in the designs should maybe not be attributed with too much significance. In order to investigate this further, the $Re = Pe = 100$ problem is optimised for a range of maximum Brinkman penalisation coefficients. Figure 5.17 shows the design and temperature distribution obtained for $\overline{\alpha} = 10^5$ in comparison with the equivalents for $\overline{\alpha} = 10^6$. Most of the designs resemble that in figure 5.17b and therefore only this is shown. It can be seen that the obtained design for $\overline{\alpha} = 10^5$ is significantly different than the $\overline{\alpha} = 10^6$ equivalent and actually performs better with respect to thermal compliance, also compared to the $Re = Pe = 10$ design. The design becomes a smooth curved surface that guides the flow over the heat sink. Two small islands of porous material can be seen in the upper left-hand side region of the design domain and when looking at the streamlines in figure 5.17d it can be seen that these obstruct the flow, collects it and forces it along the curved surface at relatively high velocities. It could be that these two islands of porous material are artifacts from the optimisation process which have not been completely removed at convergence, but it appears that they affect the flow in a manor that is beneficial for the heat transfer. It could be tested whether the islands are particularly beneficial by analysing the design without them, but this has unfortunately not been done due to time constraints. It should be noted that the design in figure 5.17b actually only uses 46% of the design domain, which means that the constraint on solid volume is not active.



(a) $\overline{\alpha} = 10^6$ **(b)** $\overline{\alpha} = 10^5$



(c) Temperature field, $\overline{\alpha} = 10^6$



(d) Temperature field, $\overline{\alpha} = 10^5$

Figure 5.17 – Comparison of optimised designs for a maximum Brinkman penalisation coefficient of $\overline{\alpha} = 10^5$ and $\overline{\alpha} = 10^6$, for the heat sink problem in a channel flow from the left. Note that the temperature range has been adjusted to show details of the temperature distribution in the fluid.
Objective function at $\overline{\alpha} = 10^6$: **(a)** $f = 1.724 \cdot 10^0$ - **(b)** $f = 1.642 \cdot 10^0$
Design iterations: **(a)** 180 - **(b)** 196

5.4 Three-dimensional heat sink - channel flow

5.4.1 Problem description

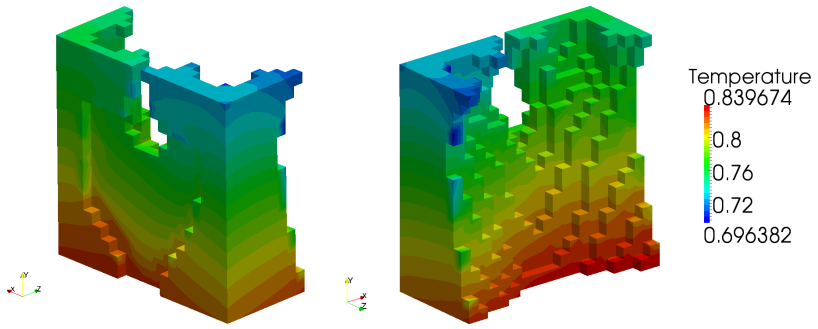
The heat sink inside a channel flow problem is now expanded to three-dimensions in order to provide a more realistic analysis of the problem. The layout and boundary conditions are the same as for the two-dimensional case, figure 5.15, but is now expanded in the out-of-plane x_3-direction. The inlet velocity profile is now parabolic in both the x_2- and x_3-directions and the channel is as wide as it is high, with the cubic design domain placed in the middle of the channel in the x_3-direction.

The problem is investigated for varying $Re = Pe$ under constant parameters, $C_k = 10^{-2}$ and $\overline{\alpha} = 10^6$.

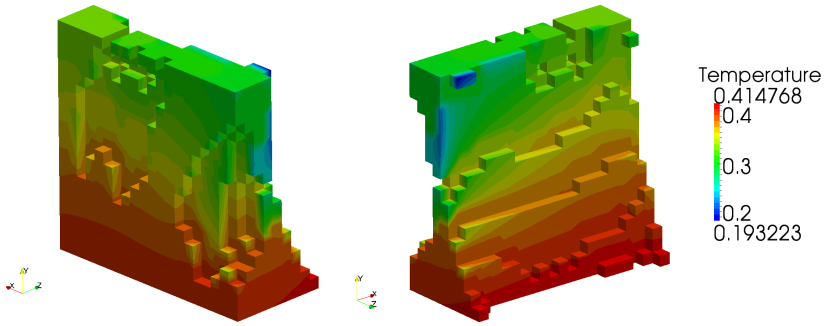
The problem is discretised with a very coarse mesh, due to the fact that currently the only available solution method is a direct factorisation and solution of the linear equation system at each nonlinear iteration. The design domain is discretised using $20 \times 20 \times 20$ cubic elements and the flow domain is discretised using elements increasing in size, the further away from the design domain they are. The flux base is discretised by $20 \times 4 \times 20$ elements with only temperature DOFs. The total number of elements is 165920, the total number of velocity and pressure DOFs is 698544 and the total number of temperature DOFs is 175518. The filter radius is set to 0.06, which is 1.2 times the element size in the design domain.

5.4.2 Varying $Re = Pe$

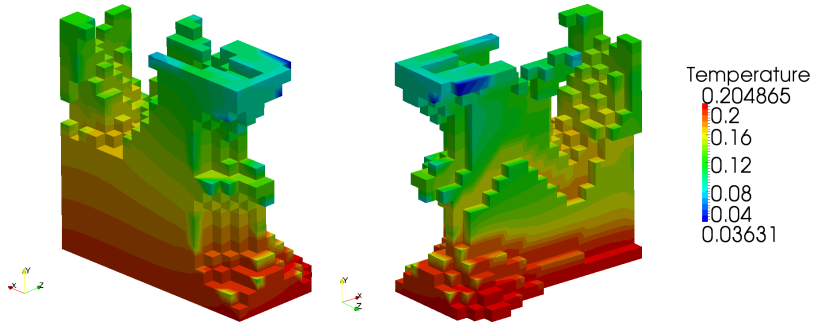
Figure 5.18 shows the obtained designs for the three-dimensional heat sink problem in a channel with flow in the positive x_1-direction (red arrow), at increasing $Re = Pe$. The figures only show elements with a design variable below a threshold of 0.1, meaning that only the "most solid" elements are visualised, and only half of the design is shown from two angles. The designs are all symmetric with respect to the midplane parallel to the flow direction. The first observation that can be made from the three-dimensional results is that the end designs look much like the ordinary straight fin heat sinks that are used for electronics cooling, in a very coarse Duplo-brick representation. The three-dimensionality opens up for the possibility of distributing material in a way that allows the fluid flow to pass through the design domain, unlike with the two-dimensional results where the flow was forced to pass over the design. This can be clearly seen in figure 5.19, where the design for $Re = Pe = 10$ is shown along with the streamlines for the fluid flow around it. Returning to figure 5.18, it is interesting to note that as $Re = Pe$ is increased, more material is placed further upstream where the cool fluid meets the heat sink, as compared to the $Re = Pe = 1$ design which is more or less quarter-symmetric. It can also be seen from figures 5.18b and 5.18c that it appears that as $Re = Pe$ is increased, the front of the "fins"



(a) $Re = Pe = 1$

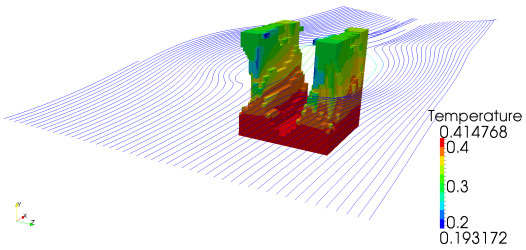


(b) $Re = Pe = 10$

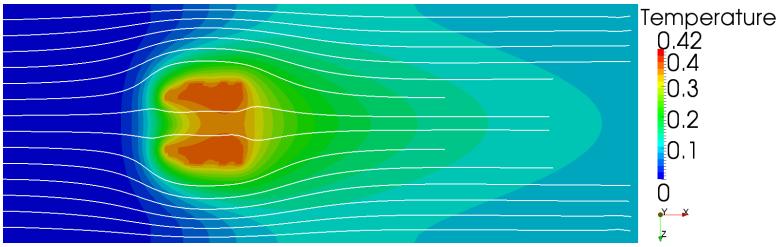


(c) $Re = Pe = 100$

Figure 5.18 – Optimised designs coloured by the corresponding temperature field for the three-dimensional heat sink problem in a channel with flow in the positive x_1-direction (red arrow), at various $Re = Pe$. Note that the temperature ranges are different and that these are threshold plots, showing elements with a design variable between 0 and 0.1.
Objective function: **(a)** $f = 3.684 \cdot 10^{0}$ - **(b)** $f = 1.818 \cdot 10^{-1}$ - **(c)** $f = 9.009 \cdot 10^{-3}$
Design iterations: **(a)** 221 - **(b)** 236 - **(c)** 260 (stopped)



(a) $Pe = 10$



(b) $Pe = 100$

Figure 5.19 – Temperature distributions and streamlines for **(a)** the optimised design shown in figure 5.18b, $Re = Pe = 10$ and **(b)** a horisontal slice through the middle of the design domain. Note that the temperature ranges are different.

become sharpened. This might be to split the flow in a favourable manor. It is likewise interesting to note that the design for $Re = Pe = 100$ is significantly more complex than the two other designs. However, the design optimised for $Re = Pe = 100$ outperforms the other two designs for all flow conditions. The reason for achieving relatively bad designs for the other flow conditions may very well be due to local minima, but no definite conclusions can be made for such a coarse discretisation. A much finer mesh should be used in order to resolve the flow field to a higher degree, maybe combined with a smaller filter radius in order to let optimal details, e.g. several fins, show themselves.

Chapter 6

Natural convection

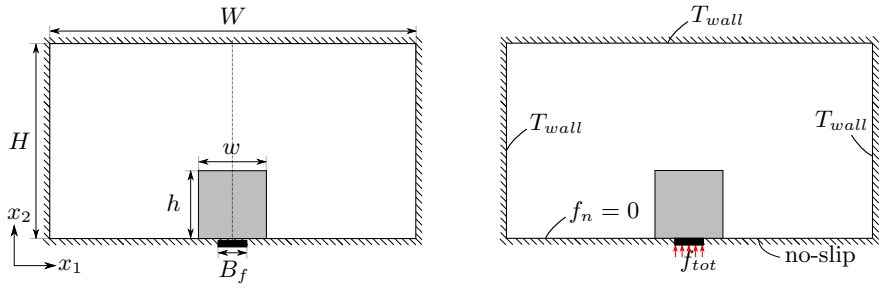
This chapter contains the optimisation results for a single problem subjected to natural convection cooling.

6.1 Two-dimensional heat sink

6.1.1 Problem description

The problem is a heat sink subjected to natural convection cooling due to surrounding cold walls. Figure 6.1 shows schematic illustations of the layout and boundary conditions for the problem. The calculation domain consists of a square design domain on top of a block of solid material that is subjected to a distributed heat flux, f_{tot}, along the bottom and a rectangular flow domain surrounding the heat sink. The upper and side walls are kept at a specified temperature, T_{wall}, and the bottom wall is thermally insulated, $f_n = 0$. All walls have no-slip conditions imposed, $u_i = 0$.

Table 6.1 lists the dimensionless quantities specifying the layout and



(a) Sizes **(b)** Boundary conditions

Figure 6.1 – Schematic illustration of the sizes and boundary conditions for the heat sink problem subjected to natural convection. Black denotes fully solid and light grey denotes the design domain.

Sizes	
Total height:	$H = 3$
Total width:	$W = 5$
Design height:	$h = 1$
Design width:	$w = 1$
Flux width:	$B_f = 0.2$

(a) Sizes

BCs	
Inlet:	$U_{in} = 1$
	$T_{in} = 0$
Outlet:	$p_{out} = 0$
Flux:	$f_{tot} = 2.2 \cdot 10^{-3}$

(b) Boundary conditions

Table 6.1 – The dimensionless quantities used for the natural convection heat sink problem shown in figure 6.1.

boundary conditions of the natural convection heat sink problem. All of the quantities specified are kept constant throughout. All spatial dimensions are relative to the initial height of the design domain, h, the flow velocities are relative to the diffusion velocity, U_{diff}, as explained in section 2.4 and the flux and temperature are relative to the scales defined in appendix E.

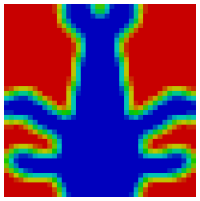
The problem is investigated for varying Ra under constant parameters, $C_k = 10^{-2}$ and $\overline{\alpha} = 10^4$, and with $Re = Pe = 1$.

The computational domain, excluding the solid flux base, is discretised using 200×120 square elements, where the design domain is discretised using 40×40. The solid flux base is discretised using 8×4 elements with only temperature DOFs. The total number of state DOFs is 243282.

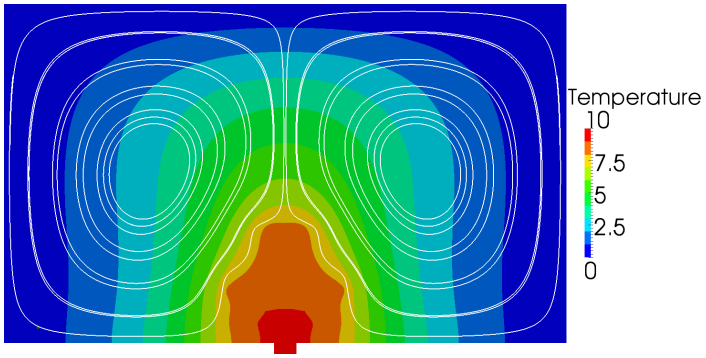
6.1.2 Varying Ra

Figure 6.2 shows the optimised designs and the corresponding temperature fields with streamlines superimposed. It can be seen that the flows are very similar, consisting of two convection cells with one on either side of the heat sink. It can be seen that as Ra is increased, the temperature field goes from a diffusion-dominated "even" distribution to a convection-dominated ploom, where the heat rises above the pointed heat sink.

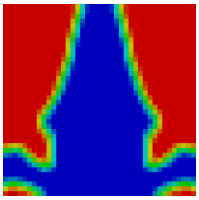
Table 6.2 shows the cross-check values for the optimised designs for the heat sink problem subjected to natural convection, shown in figure 6.2. It can be seen that all three designs perform very close for $Ra = 10$ and $Ra = 100$, but that the pointed designs, $Ra = 100$ and $Ra = 1000$, clearly perform better than the design for $Ra = 10$ at $Ra = 1000$. This is likely due to the fact that convection, and thus the flow, is not dominant at the lower Ra and the overall topology does not have as much an effect on the surrounding flow field. This is also as observed in section 5.2.



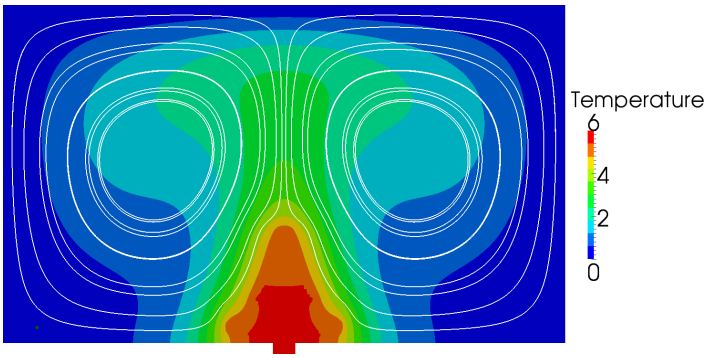
(a) $Ra = 10$



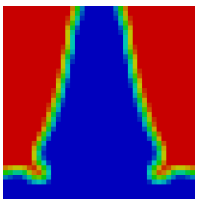
(b) Temperature field, $Ra = 10$



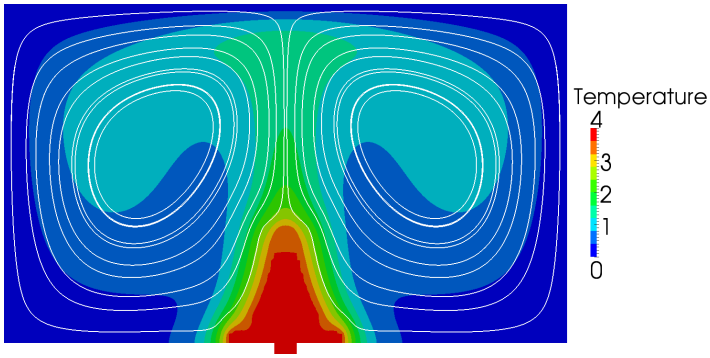
(c) $Ra = 100$



(d) Temperature field, $Ra = 100$



(e) $Ra = 1000$



(f) Temperature field, $Ra = 1000$

Figure 6.2 – Optimised designs and corresponding temperature fields for the narrow heat sink problem subject to natural convection flow at various Ra. Note that the temperature ranges are different.
Objective function: **(a)** $f = 2.045 \cdot 10^{-3}$ - **(c)** $f = 1.258 \cdot 10^{-2}$ - **(e)** $f = 8.809 \cdot 10^{-4}$
Design iterations: **(a)** 222 - **(c)** 157 - **(e)** 160

Analysis \ Optimisation	10	100	1000
10	$2.045 \cdot 10^{-3}$	$2.044 \cdot 10^{-3}$	$2.052 \cdot 10^{-3}$
100	$1.289 \cdot 10^{-3}$	$1.25786 \cdot 10^{-3}$	$1.25785 \cdot 10^{-3}$
1000	$9.184 \cdot 10^{-4}$	$8.861 \cdot 10^{-4}$	$8.809 \cdot 10^{-4}$

Table 6.2 – Cross-check of the optimised designs for the heat sink problem subjected to natural convection, shown in figure 6.2. Coloured text highlights the minimum values for a given analysis, where blue indicates values where a given optimised design performs the best for its own analysis parameters and red indicates values where this is not the case.

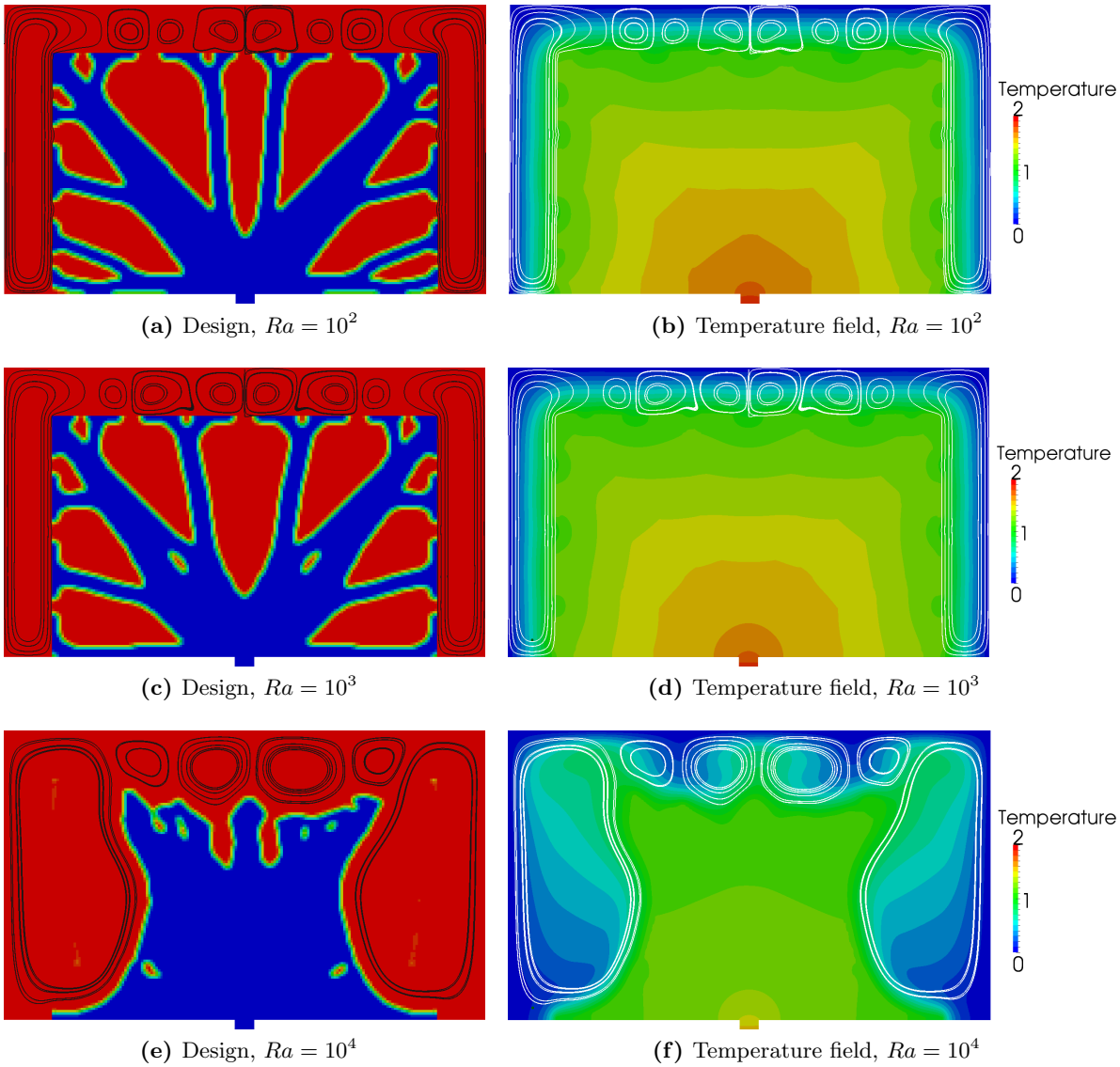
6.1.3 Enlarged design domain

As for the heat sink problem subjected to a flow from above, section 5.2, it is postulated that the reason for achieving non-optimal designs, with respect to designs for other flow conditions, is that the design domain is too small to significantly affect the overall behaviour of the surrounding flow. Therefore, the natural convection heat sink problem is investigated with a larger design domain. The design domain is enlarged to $w \times h = 4 \times 2.5$. The total number of elements and state DOFs for the computational domain remains the same but the design domain is now discretised using 160×100.

Figure 6.3 shows the optimised design, along with the corresponding temperature fields and streamlines, for the natural convection heat sink problem with an enlarged design domain. It can be seen that the designs for $Ra = 10^2$ and $Ra = 10^3$ are very similar and their performance is also very similar when cross-checked. The designs consist of conductive branches that reach out to the edges of the design domain, likely because this is as close to the cool boundary conditions that they can get. Thus, these two problems seem to be diffusion-dominated. Convection cells do appear inbetween the braches, but the temperature differences between the sides are rather moderate and thus the flow velocities are not very high.

However, the design for $Ra = 10^4$, shown in figure 6.3e, can be seen to be significantly different with a design where the material is collected more together, with no significant branches. When looking at the temperature distribution in figure 6.3f, it can be seen that it is significantly different than for the earlier cases and that thermal convection by the convection cells begin to have an importance.

Table 6.3 shows the cross-check values for the cross-check of the optimised designs for the heat sink problem subjected to natural convection with an enlarged design domain, shown in figure 6.3. As already noted, it can be seen that the performance of the designs for $Ra = 10^2$ and $Ra = 10^3$ are very similar at all Ra, whereas the design for $Ra = 10^4$ is the best for $Ra = 10^4$ but performs significantly worse than the other two designs at



(a) Design, $Ra = 10^2$ **(b)** Temperature field, $Ra = 10^2$

(c) Design, $Ra = 10^3$ **(d)** Temperature field, $Ra = 10^3$

(e) Design, $Ra = 10^4$ **(f)** Temperature field, $Ra = 10^4$

Figure 6.3 – Optimised designs and corresponding temperature fields for the heat sink problem with an enlarged design domain subject to natural convection flow at various Ra.
Objective function: **(a)** $f = 4.155 \cdot 10^{-4}$ - **(c)** $f = 4.072 \cdot 10^{-4}$ - **(e)** $f = 3.142 \cdot 10^{-4}$
Design iterations: **(a)** 226 - **(c)** 248 - **(e)** 215

		Optimisation		
		10^2	10^3	10^4
Analysis	10^2	$4.155 \cdot 10^{-4}$	$4.076 \cdot 10^{-4}$	$6.135 \cdot 10^{-4}$
	10^3	$4.152 \cdot 10^{-4}$	$4.072 \cdot 10^{-4}$	$4.769 \cdot 10^{-4}$
	10^4	$3.926 \cdot 10^{-4}$	$3.839 \cdot 10^{-4}$	$3.142 \cdot 10^{-4}$

Table 6.3 – Cross-check of the optimised designs for the heat sink problem with an enlarged design domain subjected to natural convection, shown in figure 6.3. Coloured text highlights the minimum values for a given analysis, where blue indicates values where a given optimised design performs the best for its own analysis parameters and red indicates values where this is not the case.

their flow conditions. This further indicates that the conductivity of the design dominates at the two lower Ra and when $Ra = 10^4$ the thermal convection dominates and the design thus needs to be significantly different.

It is pertinent to note the significant difficulties that arise for large Ra problems. As the convection becomes dominant, the flow and temperature fields become increasingly coupled and the problem therefore becomes increasingly nonlinear. During the topology optimisation process, the design changes significantly and the complexity of the problem therefore develops during the design history. Thus, it is argued that there is a need for a more robust solver than the current pure Newton method implementation. The design in figure 6.3e was very difficult to obtain and the final design shown was achieved by carrying over the solution from the previous design iteration as a start guess for the nonlinear solver at the next design iteration. This works reasonably well when the design changes between two iterations are reasonably small, but the drawback is also that if an analysis does not fully converge, a bad start guess is used for the next analysis. Furthermore, a bad solution for a given design iteration also gives rise to bad sensitivities, as the adjoint variables depend on the solution of the state field. The nonsymmetric details seen in figure 6.3e have likely appeared due to bad analyses during the optimisation history. Ramping of Ra has also been attempted where, for a given design iteration, the state equations are solved for progressively higher Ra until the target is met, but this did not work for the current problem. These difficulties are further discussed in section 7.2.

Part III

Discussion and conclusion

Chapter 7

Discussion

Throughout this thesis, a lot of difficulties have been encountered. Many of them have been solved or remedied, but some remain unsolved. This chapter is an attempt to give an overview of the problematic areas, what has been done and what could be done. The problematic areas are collected in three main groups and a discussion is had under each point.

7.1 Modelling

The modelling of the physical problem to be optimised is the key to an accurate and reliable end result. Without accurate modelling of the underlying physics, the topology optimisation results will be useless.

7.1.1 Stabilisation for Brinkman approach

One of the main difficulties faced, during the work for this thesis, has been the stabilised finite element formulation of the Brinkman-penalised incompressible Navier-Stokes equations. Deriving and implementing the stabilised finite element formulation for the ordinary incompressible Navier-Stokes equations is difficult and time consuming in itself, as documented in the special course report [7] written as prepatory work for this thesis. The reason for this is that there is no authoritative and well-written book on the subject. Furthermore, the Navier-Stokes equations themselves are very complicated and can exhibit outspoken nonlinear behaviour, even at relatively low Reynolds numbers.

Topology optimisation problems are problems with large variations in material properties and physical behaviour. In the context of fluid dynamics, introducing a moving immersed solid into the calculation domain poses several difficulties for the ordinary stabilised finite element formulation.

The first difficulty is the fact that the design is changed during the optimisation process and local mesh refinement cannot be used to resolve the

shear layers that form, as the boundaries and overall flow field are changed constantly. Global mesh refinement is ultimately the only option if one wishes to resolve the large solution gradients occuring at the shear layers, but when performing topology optimisation consisting of several hundred design iterations, the total time spend on solving the state equations quickly rises very fast with mesh refinement. Thus, stabilised finite elements are a viable means of getting a representative solution for relatively coarse meshes.

The second difficulty is the Brinkman approach to penalising the velocities inside of the solid parts of the domain. The Brinkman approach relatively succesfully imposes a no-slip condition along the boundary of the solid domain, but the large velocity gradients, that occur due to large material gradients, give rise to oscillations of the state field at the solid-fluid boundaries. If these large gradients occur in the streamline directions, the SUPG stabilisation method does very well to subdue the oscillations in the fluid part of the calculation domain. However, large oscillations can still occur inside the Brinkman solid at the boundaries. Furthermore, if the large gradients occur normal to the streamline directions, the SUPG stabilisation method is no longer enough and one must explore additional stabilisation such as discontinuity-capturing directional dissipation methods or crosswind diffusion methods [60, 61]. The author has tested several discontinuity-capturing methods and none of them have worked on the problem at hand, except for the discontinuity capturing methods for the temperature field [10, 20], which helped significantly for large Péclet number flows. It is postulated that the reason, for why the flow stabilising discontinuity-capturing methods did not work convincingly, is due to the fact that these were developed for use in pure fluid domains. The velocities inside of the Brinkman solid are much smaller than in the fluid domain and this, coupled with the fact that the ordinary stabilisation terms are many orders of magnitude lower the Brinkman penalisation terms, basically leads to the discontinuity-capturing methods having no effect inside of the Brinkman solid.

The discussion above leads to the following point, that if stabilised finite element formulations are to be used together with the Brinkman approach to topology optimisation, the development of specialised stabilisation methods for large material gradients could be an option to mitigate the oscillations that occur in the state field. One could argue that as long as the oscillations are contained within the Brinkman solid, then it does not matter with respect to the solution of the state field, as the velocities are still low. However, as it has been reported by several authors [38, 48] for fluid flow topology optimisation, oscillations in the state field at the solid-fluid interfaces can lead to problems with the adjoint fields and subsequently the sensitivities.

Furthermore, the standard PSPG stabilisation within the Brinkman solid, with the stabilisation parameters defined in appendix A.7, suffered from significant convergence issues for large $\overline{\alpha}$. The oscillations at the solid-fluid interfaces travelled into the porous region and were not localised at the in-

terfaces. Furthermore, the pressure distribution was also oscillatory and exhibited large variations at the solid-fluid interfaces. Both of these issues were mitigated by making the stabilisation parameter, τ_{PS}, dependent on the porosity, as explained in appendix A.7, which was inspired by [15, 39].

7.1.2 Brinkman approach for natural convection

The applicability of the Brinkman approach for natural convection flow problems should also be investigated further in order to provide a good foundation for expanding the use of the approach to these kind of problems. Unfortunately there has not been time for the author to conduct thorough investigations for the introduction of the Brinkman penalisation term to the natural convection flow problems, but a quick investigation was performed as mentioned in section 4.2.3. There exists several papers in the literature on the subject of natural convection in and around porous medium which could help aid this further investigation.

With respect to whether the Boussinesq forcing term should be made dependent on the design variable, and thus zero inside the solid elements, it is the authors opinion that there exists a fundamental question about whether the parts of the design domain modelled using the Brinkman approach should be treated as truly solid or as the semi-permeable material that it actually is.

7.2 Solving state equations

The solution of the nonlinear state equation system has also possed a few difficulties throughout the work for this thesis. Generally these can be boiled down to the following chain of relations: Interesting problems are the realistic problems, and realistic problems are three-dimensional and nonlinear, and three-dimensional and nonlinear problems are the difficult problems to solve.

7.2.1 Robust nonlinear solver

The simple nonlinear Newton solver, as described in section 3.3, has been adequate for many of the problems studied during this thesis. Most often the full Newton step could be taken leading to rapid convergence, but as the nonlinearity of the problems increases, the fraction taken of the Newton step had to be decreased in order to converge. Due to the fact that the design, and thus the complexity of the flow problem, can change significantly during a topology optimisation process, often a small steplength parameter had to be chosen in order to provide the solver with some robustness against increasing nonlinearity during the design process.

There is no doubt that it is not adequate to have a constant steplength parameter throughout the optimisation process, as one has to chose a smaller steplength in order to be on the safe side for very nonlinear problems. This leads to more nonlinear iterations per state field evaluation and thus quickly increases the total time taken for the entire optimisation process. Furthermore, by taking a constant fraction of the Newton step throughout the nonlinear iterations, one looses out on the full quadratic convergence of Newtons method when one approaches the solution.

During the work for this thesis, both residual- and error-based affine contravariant adaptive schemes were tested [25] in order to improve the robustness of the nonlinear solver, but none of them were successful in providing a better solver. In hindsight, this is likely due to the fact that scaling of functions and variables, as described in e.g. [19], was not performed on the equation system. It is postulated that combining scaling with the adaptive schemes can provide a robust nonlinear solver suitable for topology optimisation of strongly convective flow problems, both forced and natural. Unfortunately, there has not been time to test this.

7.2.2 Iterative solvers for three-dimensional problems

It is well known that direct solution methods are not efficient for three-dimensional problems and that iterative solvers are the only efficient way to solve the equation systems that arise. Partly due to the large memory requirements for the factorisation, but also due to the poor parallelisation properties of direct methods. However, in order to have an efficient iterative solver, one needs a good preconditioner. Once again, taking the Navier-Stokes equations, which yield a non-symmetric state matrix, and mixing it with topology optimisation, which yields large contrasts in coefficients, leads to problems. It is possible to use rather simple preconditioners, such as the Jacobi or diagonal preconditioner, for solving the stabilised finite element equations for coupled convection problems [63], but it is very likely that this would fail for topology optimisation problems exhibiting large contrasts in coefficients. Also, it should be noted that it might be tolerable for a single analysis to use a large amount of iterations within the iterative solver, but for topology optimisation problems with hundreds of design iterations and thus state evaluations, a highly efficient solution method is needed. Work is currently being done within the research group on efficient multigrid preconditioners, but unfortunately this has not been ready for use during this thesis.

7.3 Physical relevance of results

As already stated in the above section, the interesting problems are the problems that are difficult to solve. The results presented in this thesis are

thus mainly two-dimensional and at low Reynolds and Rayleigh numbers.

7.3.1 Two-dimensional results

The physical relevance of the two-dimensional results for the heat sink problems, presented in sections 5.2 and 5.3 can be questioned. The results are modelled as two-dimensional, which means that the fluid can only move in-plane. This poses difficulties for the design of heat sinks, because the fluid cannot move inbetween or past the members that are formed. Not many realistic applications of heat sinks would have this limitation, but as often is in general engineering and research, one must simplify the problems significantly in order to analyse them. It should be noted, that although the physical relevance may be debateable, the problems still give rise to many valid and useful observations about the topology optimisation of coupled convection problems.

7.3.2 Low Rayleigh number

The results presented in section 6.1 are for relatively low Rayleigh numbers and it is questionable whether the problems are physically realistic. When a fluid is heated from below, from for instance a heat sink, with a small heat flux, the heat simply diffuses through the fluid without causing the fluid to flow. When the heat flux is increased, a critical Rayleigh number will be encountered and the system goes from a stable state to the natural convection state, where the fluid begins to flow due to the bouyancy forces overpowering the restoring viscous forces.

Chapter 8

Conclusion

It can hereby be concluded that the aim of the project has been fulfilled. In this thesis, topology optimisation has been extended and applied to coupled convection problems. Several conjugate heat transfer problems have been investigated as well as a single natural convection problem.

Although the presented results are very simple and remain academic in nature, it is envisioned that by further development the methodology, presented in this thesis, can be used to optimise realistic industrial problems such as the cooling of combustion engines and electronics. Furthermore, the presented work is seen as contributing new research to the field of topology optimisation for multiphysics problems. The topology optimisation of conjugate heat transfer problems is not very well documented in the literature and to the authors knowledge, topology optimisation has not been applied to natural convection problems before.

8.1 Further work

The work for the thesis has definitely not been without difficulties as described in chapter 7. The problem areas outlined here are also the main areas where further research and investigation is needed in the authors opinion:

- Specialised stabilisation methods taking the large gradients in material properties into account could be beneficial for topology optimisation of fluid flow problems.
- Thorough investigations are needed into the validity and behaviour of the Brinkman approach for natural convection, before topology optimisation can be applied to these problems with confidence.
- A robust nonlinear solver is seen as absolutely essential for the further development and application of topology optimisation to nonlinear flow problems, both forced and natural.
- Efficient iterative methods for the solution of three-dimensional problems will allow topology optimisation to be used for realistic and interesting problems.

Part IV

Appendix

Appendix A

FEM element matrices

The following is a slightly rewritten excerpt, with additions concerning the Brinkman terms, from the special course report [7] conducted as prepatory work for this thesis

The PSPG- and SUPG-stabilised Galerkin finite element method results in the following discretised system of equations, in its most general form including the Boussinesq coupling and Brinkman term:

$$\begin{bmatrix} \mathbf{C}(\mathbf{u}) + \mathbf{K} + \boldsymbol{\alpha} + \mathbf{L}_\delta(\mathbf{u}) + \mathbf{L}_{\alpha\delta}(\mathbf{u}) & \mathbf{G} + \mathbf{A}_\delta(\mathbf{u}) & \mathbf{B} + \mathbf{Q}_\delta(\mathbf{u}) \\ \mathbf{G}^T + \mathbf{L}_\epsilon + \mathbf{L}_{\alpha\epsilon} & \mathbf{A}_\epsilon & \mathbf{Q}_\epsilon \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_t(\mathbf{u}) + \mathbf{K}_t + \mathbf{Q}_\zeta(\mathbf{u}) \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{p} \\ \mathbf{t} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} + \mathbf{r}_\delta(\mathbf{u}) \\ \mathbf{r}_\epsilon \\ \mathbf{f}_t + \mathbf{r}_\zeta(\mathbf{u}) \end{Bmatrix} \tag{3.4}$$

where subscript ϵ denotes PSPG stabilisation, subscript δ denotes SUPG stabilisation, subscript ζ denotes thermal SUPG stabilisation and all of the matrices and vectors are as defined in the following sections. It should be noted, that only the direct dependence on the velocity field is noted above. The implicit dependence of the stabilisation parameters on the velocity and temperature fields has been left out for simplicity, but the full nonlinear dependence is shown in appendix B. The global vectors and matrices are all obtained from the finite element assembly of all the element equivalents:

$$\square = \mathop{\mathsf{A}}_{e=1}^{n_\mathrm{e}} \square^e \tag{A.1}$$

where $\mathsf{A}_{e=1}^{n_\mathrm{e}}$ is the finite element assembly operator. The sizes of the matrices and vectors are given for three-dimensional hexahedral elements as shown in figure A.1.

A.1 Convection-diffusion

Convection matrix

The element convection matrix is given by:

$$\underset{8\times8}{\mathbf{C}_t^e} = \int_{\Omega^e} \underset{8\times1}{\mathbf{N}_t^{\mathrm{T}}} \underset{1\times8}{\mathbf{D}_t}\, dV \tag{A.2}$$

where $\mathbf{N}_t$ is the temperature shape function matrix and $\mathbf{D}_t$ is the thermal convection interpolation matrix.
The temperature shape function matrix is defined as:

$$\mathbf{N}_t = \begin{bmatrix} N_{t1} & N_{t2} & N_{t3} & N_{t4} & N_{t5} & N_{t6} & N_{t7} & N_{t8} \end{bmatrix} \tag{A.3}$$

where N_{ti} is the temperature shape function for node number i.
The thermal convection interpolation matrix is defined as:

$$\underset{1\times8}{\mathbf{D}_t} = \underset{1\times3}{\mathbf{u}_x{}^{\mathrm{T}}} \underset{3\times8}{\mathbf{B}_t} \tag{A.4}$$

where $\mathbf{u}_x$ is the spatially varying velocity vector and $\mathbf{B}_t$ is the flux-temperature matrix.
The positional velocity is interpolated from the given velocity field as follows:

$$\underset{3\times1}{\mathbf{u}_x} = \underset{3\times24}{\mathbf{N}_u}\, \underset{24\times1}{\mathbf{u}_e} \tag{A.5}$$

where $\mathbf{N}_u$ is the velocity shape function matrix, defined later in equation (A.16), and $\mathbf{u}_e$ is the velocity vector for the given element.
The flux-temperature matrix is defined as:

$$\underset{3\times8}{\mathbf{B}_t} = \underset{3\times1}{\boldsymbol{\nabla}}\, \underset{1\times8}{\mathbf{N}_t} \tag{A.6}$$

where the divergence operator vector is defined as:

$$\boldsymbol{\nabla} = \left\{ \frac{\partial}{\partial x_1} \quad \frac{\partial}{\partial x_2} \quad \frac{\partial}{\partial x_3} \right\}^{\mathrm{T}} \tag{A.7}$$

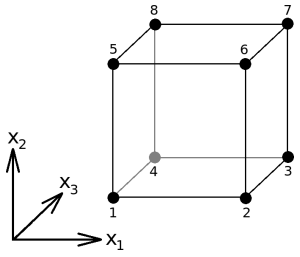


Figure A.1 – An arbitrary three-dimensional hexahedral finite element.

Diffusivity matrix

The element diffusivity matrix is given by:

$$\underset{8\times8}{\mathbf{K}_t^e} = \int_{\Omega^e} \underset{8\times3}{\mathbf{B}_t}^{\mathrm{T}} \underset{3\times3}{\mathbf{P}_e} \underset{3\times8}{\mathbf{B}_t}\, dV \tag{A.8}$$

where $\mathbf{B}_t$ is the flux-temperature matrix, as defined in equation (A.6), and

$$\underset{3\times3}{\mathbf{P}_e} = \frac{1}{Pe}\underset{3\times3}{\mathbf{I}} \tag{A.9}$$

is the Péclet matrix.

Energy flux vector

The energy flux vector is given by two contributions, one from the Neumann boundary conditions on temperature flux and one from the volumetric heat source term:

$$\underset{8\times1}{\mathbf{f}_t^e} = \underset{8\times1}{\mathbf{f}_f^e} + \underset{8\times1}{\mathbf{f}_{s_t}^e} \tag{A.10}$$

The element temperature flux boundary contribution is given by:

$$\underset{8\times1}{\mathbf{f}_f^e} = \sum_{i=1}^{N_{Neu}^e} \int_{\Gamma_i^e} \frac{1}{Pe} \underset{8\times1}{\mathbf{N}_t}^{\mathrm{T}} f_n^{e,i}\, dS \tag{A.11}$$

where $\mathbf{N}_t$ is the temperature shape function matrix, as defined in equation (A.3), $f_n^{e,i}$ is the temperature flux on side i of element e, N_{Neu}^e is the number of sides of element e that have the Neumann boundary condition applied to them and Γ_i^e denotes the surface of side i of element e.

The element volumetric heat source contribution is given by:

$$\underset{8\times1}{\mathbf{f}_{s_t}^e} = \int_{\Omega^e} \underset{8\times1}{\mathbf{N}_t}^{\mathrm{T}} s_t^e\, dS \tag{A.12}$$

where $\mathbf{N}_t$ is the temperature shape function matrix and s_t^e is the volumetric heat source for element e.

A.2 Fluid flow

Convection matrix

The element convection matrix is given by:

$$\underset{24\times24}{\mathbf{C}^e} = \int_{\Omega^e} \underset{24\times3}{\mathbf{N}_u}^{\mathrm{T}} \underset{3\times24}{\mathbf{D}_u}\, dV \tag{A.13}$$

where $\mathbf{N}_u$ is the velocity shape function matrix and $\mathbf{D}_u$ is the convection interpolation matrix. The convection interpolation matrix is defined as:

$$\underset{3\times24}{\mathbf{D}_u} = \underset{3\times3}{\mathbf{I}_{u\nabla}}\ \underset{3\times24}{\mathbf{N}_u} \tag{A.14}$$

where $\mathbf{N}_u$ is the velocity shape function matrix and $\mathbf{I}_{u\nabla}$ is the convective operator matrix, which is defined as having the convective operator along the diagonal and zeroes elsewhere. The convective operator matrix can be expressed as:

$$\underset{3\times3}{\mathbf{I}_{u\nabla}} = \begin{bmatrix} {\mathbf{u}_x}^{\mathrm{T}}\boldsymbol{\nabla} & 0 & 0 \\ 0 & {\mathbf{u}_x}^{\mathrm{T}}\boldsymbol{\nabla} & 0 \\ 0 & 0 & {\mathbf{u}_x}^{\mathrm{T}}\boldsymbol{\nabla} \end{bmatrix} \tag{A.15}$$

where $\mathbf{u}_x$ is the spatially varying velocity vector, as defined in equation (A.5), and $\boldsymbol{\nabla}$ is the divergence vector, as defined in equation (A.7).
The velocity shape function matrix, $\mathbf{N}_u$, is defined as follows:

$$\mathbf{N}_u = \begin{bmatrix} N_{u1} & 0 & 0 & N_{u2} & 0 & \cdots & 0 & 0 \\ 0 & N_{u1} & 0 & 0 & N_{u2} & \cdots & N_{u8} & 0 \\ 0 & 0 & N_{u1} & 0 & 0 & \cdots & 0 & N_{u8} \end{bmatrix} \tag{A.16}$$

where N_{ui} is the velocity shape function for node number i.

Viscosity/diffusivity matrix

The element viscosity matrix is given by:

$$\underset{24\times24}{\mathbf{K}^e} = \int_{\Omega^e} \underset{24\times6}{{\mathbf{B}_u}^{\mathrm{T}}}\ \underset{6\times6}{\mathbf{R}_e}\ \underset{6\times24}{\mathbf{B}_u}\ dV \tag{A.17}$$

where $\mathbf{B}_u$ is the strainrate-velocity matrix and $\mathbf{R}_e$ is the Reynolds matrix. The strainrate-velocity matrix is given by:

$$\underset{6\times24}{\mathbf{B}_u} = \underset{6\times3}{\boldsymbol{\partial}_u}\ \underset{3\times24}{\mathbf{N}_u} \tag{A.18}$$

where the differentiation matrix is defined as:

$$\boldsymbol{\partial}_u = \begin{bmatrix} \frac{\partial}{\partial x_1} & 0 & 0 & \frac{\partial}{\partial x_2} & 0 & \frac{\partial}{\partial x_3} \\ 0 & \frac{\partial}{\partial x_2} & 0 & \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_3} & 0 \\ 0 & 0 & \frac{\partial}{\partial x_3} & 0 & \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} \end{bmatrix}^{\mathrm{T}} \tag{A.19}$$

The Reynolds matrix is defined as:

$$\mathbf{R}_e = \frac{1}{Re}\begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.20}$$

Pressure-coupling matrix

The element pressure-velocity coupling matrix is given by:

$$\underset{24\times 8}{\mathbf{G}^e} = \int_{\Omega^e} \underset{24\times 3}{\mathbf{N}_u}^{\mathrm{T}} \underset{3\times 8}{\mathbf{B}_p}\, dV \tag{A.21}$$

where $\mathbf{N}_u$ is the velocity shape function matrix, as defined in equation (A.16), and $\mathbf{B}_p$ is the pressure-gradient matrix. The pressure-gradient matrix is defined as:

$$\underset{3\times 8}{\mathbf{B}_p} = \underset{3\times 1}{\boldsymbol{\nabla}} \underset{1\times 8}{\mathbf{N}_p} \tag{A.22}$$

where $\boldsymbol{\nabla}$ is the divergence operator vector, as defined in equation (A.7), and $\mathbf{N}_p$ is the pressure shape function matrix defined as:

$$\mathbf{N}_p = \begin{bmatrix} N_{p1} & N_{p2} & N_{p3} & N_{p4} & N_{p5} & N_{p6} & N_{p7} & N_{p8} \end{bmatrix} \tag{A.23}$$

where N_{pi} is the pressure shape function for node number i.

Boussinesq coupling matrix

The element Boussinesq coupling matrix is given by:

$$\underset{24\times 8}{\mathbf{B}^e} = \int_{\Omega^e} \frac{Ra}{PeRe} \underset{24\times 3}{\mathbf{N}_u}^{\mathrm{T}} \underset{3\times 1}{\mathbf{e}_g} \underset{1\times 8}{\mathbf{N}_t}\, dV \tag{A.24}$$

where $\mathbf{N}_u$ is the velocity shape function matrix, defined in equation (A.16), $\mathbf{e}_g$ is the unit vector in the gravitational direction and $\mathbf{N}_t$ is the temperature shape function matrix, defined in equation (A.3).

Momentum forcing vector

The momentum forcing vector is given by two contributions, one from the Neumann boundary conditions on surface traction and one from the volumetric source term:

$$\underset{24\times 1}{\mathbf{f}^e} = \underset{24\times 1}{\mathbf{f}_h^e} + \underset{24\times 1}{\mathbf{f}_s^e} \tag{A.25}$$

The element surface traction boundary contribution is given by:

$$\underset{24\times 1}{\mathbf{f}_h^e} = \sum_{i=1}^{N_{Neu}^e} \int_{\Gamma_i^e} \underset{24\times 3}{\mathbf{N}_u}^{\mathrm{T}} \underset{3\times 1}{\mathbf{h}_i^e}\, dS \tag{A.26}$$

where $\mathbf{N}_u$ is the velocity shape function matrix, as defined in equation (A.16), and $\mathbf{h}_i^e$ is the surface traction on side i of element e.

The element volumetric source contribution is given by:

$$\underset{24\times 1}{\mathbf{f}_s^e} = \int_{\Omega^e} \underset{24\times 3}{\mathbf{N}_u}^{\mathrm{T}} \underset{3\times 1}{\mathbf{s}^e}\, dV \tag{A.27}$$

where $\mathbf{N}_u$ is the velocity shape function matrix, as defined in equation (A.16), and $\mathbf{s}^e$ is the volumetric source term vector for element e.

A special case of the volumetric source term, is when only gravity forcing is present. In this case, the source term vector can be expressed based on the Froude number and the unit vector in the gravity direction:

$$\underset{3\times1}{\mathbf{s}_g} = \frac{1}{Fr^2}\underset{3\times1}{\mathbf{e}_g} \tag{A.28}$$

where usually $\mathbf{e}_g = \begin{Bmatrix} 0 & -1 & 0 \end{Bmatrix}^{\mathrm{T}}$ as gravity often is defined as acting in the negative x_2-direction. The Froude number is defined as:

$$Fr = \frac{U}{\sqrt{gL}} \tag{A.29}$$

and describes the ratio between the characteristic velocity and a gravitational wave velocity scale. This in turn leads to the gravitational forcing vector:

$$\underset{24\times1}{\mathbf{f}_g^e} = \int_{\Omega^e} \frac{1}{Fr^2}\underset{24\times3}{\mathbf{N}_u}^{\mathrm{T}}\underset{3\times1}{\mathbf{e}_g}\, dV \tag{A.30}$$

where $\mathbf{N}_u$ is the velocity shape function matrix, as defined in equation (A.16), and $\mathbf{e}_g$ is the unit vector in the gravitational direction as defined above. Due to gravity being a conservative force, it can be represented as the gradient of a scalar quantity, namely:

$$\frac{1}{Fr^2}\mathbf{e}_g = \boldsymbol{\nabla}\left(-\frac{1}{Fr^2}x_2\right) \tag{A.31}$$

with gravity acting in the negative x_2-direction. The gravity term can therefore be included, as it needs to be for the Boussinesq approximation later, by modifying the pressure to include the gravity body force:

$$-\boldsymbol{\nabla}p + \frac{1}{Fr^2}\mathbf{e}_g = -\boldsymbol{\nabla}p + \boldsymbol{\nabla}\left(-\frac{1}{Fr^2}x_2\right) = -\boldsymbol{\nabla}\left(p + \frac{1}{Fr^2}x_2\right) = -\boldsymbol{\nabla}p_m \tag{A.32}$$

where the modified pressure contains the "gravitational head", $p_m = p + \frac{1}{Fr^2}x_2$.

A.3 Brinkman term

The element Brinkman matrix is given by:

$$\underset{24\times24}{\boldsymbol{\alpha}^e} = \int_{\Omega^e} \alpha_e \underset{24\times3}{\mathbf{N}_u}^{\mathrm{T}}\underset{3\times24}{\mathbf{N}_u}\, dV \tag{A.33}$$

where $\mathbf{N}_u$ is the velocity shape function matrix, defined in equation (A.16), and α_e is the element impermeability.

A.4 PSPG stabilisation

Convection matrix

The element PSPG convection matrix is given by:

$$\underset{8\times 24}{\mathbf{L}_{\epsilon}^{e}} = -\int_{\Omega^e} \tau_{PS} \underset{8\times 3}{\mathbf{B}_p{}^{\mathrm{T}}} \underset{3\times 24}{\mathbf{D}_u}\, dV \tag{A.34}$$

where τ_{PS} is the stabilisation parameter, $\mathbf{B}_p$ is the pressure-gradient matrix, defined in equation (A.22), and $\mathbf{D}_u$ is the convection interpolation matrix, defined in equation (A.14).

Pressure matrix

The element PSPG pressure matrix is given by:

$$\underset{8\times 8}{\mathbf{A}_{\epsilon}^{e}} = -\int_{\Omega^e} \tau_{PS} \underset{8\times 3}{\mathbf{B}_p{}^{\mathrm{T}}} \underset{3\times 8}{\mathbf{B}_p}\, dV \tag{A.35}$$

where τ_{PS} is the stabilisation parameter and $\mathbf{B}_p$ is the pressure-gradient matrix, defined in equation (A.22).

Source vector

The element PSPG source term vector is given by:

$$\underset{8\times 1}{\mathbf{r}_{\epsilon}^{e}} = -\int_{\Omega^e} \tau_{PS} \underset{8\times 3}{\mathbf{B}_p{}^{\mathrm{T}}} \underset{3\times 1}{\mathbf{s}}\, dV \tag{A.36}$$

where τ_{PS} is the stabilisation parameter, $\mathbf{B}_p$ is the pressure-gradient matrix, defined in equation (A.22), and $\mathbf{s}$ is the source term vector.

Boussinesq matrix

The PSPG Boussinesq stabilisation matrix is defined as:

$$\underset{8\times 8}{\mathbf{Q}_{\epsilon}^{e}} = -\int_{\Omega^e} \tau_{PS} \frac{Ra}{PeRe} \underset{8\times 3}{\mathbf{B}_p{}^{\mathrm{T}}} \underset{3\times 1}{\mathbf{e}_g} \underset{1\times 8}{\mathbf{N}_t}\, dV \tag{A.37}$$

where τ_{PS} is the stabilisation parameter, $\mathbf{B}_p$ is the pressure-gradient matrix, defined in equation (A.22), $\mathbf{e}_g$ is the unit vector in the gravitational direction and $\mathbf{N}_t$ is the temperature shape function matrix, defined in equation (A.3).

Brinkman matrix

The PSPG Brinkman stabilisation matrix is defined as:

$$\underset{8\times24}{\mathbf{L}^e_{\alpha\epsilon}} = -\int_{\Omega^e} \tau_{PS}\alpha_e \underset{8\times3}{\mathbf{B}_p{}^{\mathrm{T}}}\ \underset{3\times24}{\mathbf{N}_u}\, dV \tag{A.38}$$

where τ_{PS} is the stabilisation parameter, $\mathbf{B}_p$ is the pressure-gradient matrix, defined in equation (A.22), $\mathbf{N}_u$ is the velocity shape function matrix, defined in equation (A.16) and α_e is the element impermeability.

A.5 SUPG stabilisation for flow

Convection matrix

The element SUPG convection matrix is given by:

$$\underset{24\times24}{\mathbf{L}^e_{\delta}} = \int_{\Omega^e} \tau_{SU} \underset{24\times3}{\mathbf{D}_u{}^{\mathrm{T}}}\ \underset{3\times24}{\mathbf{D}_u}\, dV \tag{A.39}$$

where τ_{SU} is the stabilisation parameter and $\mathbf{D}_u$ is the convection interpolation matrix, defined in equation (A.14).

Pressure matrix

The element SUPG pressure matrix is given by:

$$\underset{24\times8}{\mathbf{A}^e_{\delta}} = \int_{\Omega^e} \tau_{SU} \underset{24\times3}{\mathbf{D}_u{}^{\mathrm{T}}}\ \underset{3\times8}{\mathbf{B}_p}\, dV \tag{A.40}$$

where τ_{SU} is the stabilisation parameter, $\mathbf{D}_u$ is the convection interpolation matrix, defined in equation (A.14), and $\mathbf{B}_p$ is the pressure-gradient matrix, defined in equation (A.22).

Source vector

The element SUPG source term vector is given by:

$$\underset{24\times1}{\mathbf{r}^e_{\delta}} = \int_{\Omega^e} \tau_{SU} \underset{24\times3}{\mathbf{D}_u{}^{\mathrm{T}}}\ \underset{3\times1}{\mathbf{s}}\, dV \tag{A.41}$$

where τ_{SU} is the stabilisation parameter, $\mathbf{D}_u$ is the convection interpolation matrix, defined in equation (A.14), and $\mathbf{s}$ is the source term vector.

Boussinesq matrix

The SUPG Boussinesq stabilisation matrix is defined as:

$$\underset{24\times 8}{\mathbf{Q}^e_\delta} = \int_{\Omega^e} \tau_{SU} \frac{Ra}{PeRe} \underset{24\times 3}{\mathbf{D}_u{}^{\mathrm{T}}} \underset{3\times 1}{\mathbf{e}_g} \underset{1\times 8}{\mathbf{N}_t}\, dV \tag{A.42}$$

where τ_{SU} is the stabilisation parameter, $\mathbf{D}_u$ is the convection interpolation matrix, defined in equation (A.14), $\mathbf{e}_g$ is the unit vector in the gravitational direction and $\mathbf{N}_t$ is the temperature shape function matrix, defined in equation (A.3).

Brinkman matrix

The SUPG Brinkman stabilisation matrix is defined as:

$$\underset{24\times 24}{\mathbf{L}^e_{\alpha\delta}} = \int_{\Omega^e} \tau_{SU} \alpha_e \underset{24\times 3}{\mathbf{D}_u{}^{\mathrm{T}}} \underset{3\times 24}{\mathbf{N}_u}\, dV \tag{A.43}$$

where τ_{SU} is the stabilisation parameterr, $\mathbf{D}_u$ is the convection interpolation matrix, defined in equation (A.14), $\mathbf{N}_u$ is the velocity shape function matrix, defined in equation (A.16), and α_e is the element impermeability.

A.6 SUPG stabilisation for temperature

Convection matrix

The element SUPG thermal convection matrix is given by:

$$\underset{8\times 8}{\mathbf{Q}^e_\zeta} = \int_{\Omega^e} \tau_{SU_T} \underset{8\times 1}{\mathbf{D}_t{}^{\mathrm{T}}} \underset{1\times 8}{\mathbf{D}_t}\, dV \tag{A.44}$$

where τ_{SU_T} is the stabilisation parameter and $\mathbf{D}_t$ is the thermal convection interpolation matrix, defined in equation (A.4).

Source vector

The element SUPG thermal source term vector is given by:

$$\underset{8\times 1}{\mathbf{r}^e_\zeta} = \int_{\Omega^e} \tau_{SU_T} \underset{8\times 1}{\mathbf{D}_t{}^{\mathrm{T}}} s^e_t\, dV \tag{A.45}$$

where τ_{SU_T} is the stabilisation parameter, $\mathbf{D}_t$ is the thermal convection interpolation matrix, defined in equation (A.4), and s^e_t is the thermal source term.

A.7 Stabilisation parameters

Flow

The current implementation is based on the stabilisation parameters defined in [63] which are so-called UGN-based stabilisation parameters. The stabilisation parameters are defined as follows:

$$\tau_{PS} = \tau_{SU} = \left(\frac{1}{{\tau_{SUGN1}}^r} + \frac{1}{{\tau_{SUGN3}}^r} \right)^{-1/r} \tag{A.46}$$

which is an approximate min-function that switches between the two parameters:

$$\tau_{SUGN1} = \frac{h_{UGN}}{2\|\mathbf{u}^h\|} \tag{A.47a}$$

$$\tau_{SUGN3} = \frac{{h_{RGN}}^2 Re}{4} \tag{A.47b}$$

based on the switching parameter, r, which is usually set to 2. The length-scales used in the above parameters are defined as:

$$h_{UGN} = 2\|\mathbf{u}^h\| \left(\sum_{a=1}^{n_{\text{en}}} |\mathbf{u}^h \cdot \boldsymbol{\nabla} N_{ua}| \right)^{-1} \tag{A.48a}$$

$$h_{RGN} = 2 \left(\sum_{a=1}^{n_{\text{en}}} |\mathbf{r} \cdot \boldsymbol{\nabla} N_{ua}| \right)^{-1} \tag{A.48b}$$

where n_{en} is the number of nodes per element and $\mathbf{r}$ is a unit vector defined in the velocity-gradient direction:

$$\mathbf{r} = \frac{\boldsymbol{\nabla}\|\mathbf{u}^h\|}{\|\boldsymbol{\nabla}\|\mathbf{u}^h\|\|} \tag{A.49}$$

The stabilisation parameters are taken to be constant within each element, so the expressions above are evaluated at the element centres. This is an approximation to the full stabilisation parameters based on element matrices and vectors, as in [58], which has been simplified based on the assumption of using a single integration point to evaluate the stabilisation parameters as hinted in [59]. All norms in this section are defined to be the 2-norm.

For the Brinkman penalised elements, the standard PSPG stabilisation, with the stabilisation parameters defined as above, suffered from significant convergence issues. The oscillations at the solid-fluid interfaces travelled into the porous region and were not as localised at the interfaces. Furthermore, the pressure distribution was also oscillatory and exhibited large variations

at the solid-fluid interfaces. Inspired by the papers by Masud [39] and Braack et al. [15], both of these issues were mitigated by making the stabilisation parameters dependent on the porosity as follows:

$$\tau_{PS} = \tau_{SU} = \left(\frac{1}{{\tau_{SUGN1}}^r} + \frac{1}{{\tau_{SUGN3}}^r} + {\alpha_e}^r \right)^{-1/r} \tag{A.50}$$

where α_e is the impermeability for the given element. This makes sure that the stabilisation takes the reaction/porosity-dominance into account when the impermeability is large.

Thermal

The calculation of the stabilisation parameter for the temperature equation is basically the same as explained for fluid flow, with small changes to a few of the definitions. The temperature stabilisation parameter is defined as:

$$\tau_{SU_T} = \left(\frac{1}{{\tau_{SUGN1_T}}^r} + \frac{1}{{\tau_{SUGN3_T}}^r} \right)^{-1/r} \tag{A.51}$$

where:

$$\tau_{SUGN1_T} = \tau_{SUGN1} \tag{A.52a}$$

$$\tau_{SUGN3_T} = \frac{{h_{RGN_T}}^2 Pe}{4} \tag{A.52b}$$

The length-scale used above is defined as:

$$h_{RGN_T} = 2 \left(\sum_{a=1}^{n_{\text{en}}} |\mathbf{r}_T \cdot \boldsymbol{\nabla} N_{ta}| \right)^{-1} \tag{A.53}$$

where n_{en} is the number of nodes per element and $\mathbf{r}_T$ is a unit vector defined in the temperature-gradient direction:

$$\mathbf{r}_T = \frac{\boldsymbol{\nabla} T^h}{\|\boldsymbol{\nabla} T^h\|} \tag{A.54}$$

Appendix B

Tangent system matrix

B.1 Residual

The residual for the discretised nonlinear system of equations is defined as follows:

$$\underset{n_s \times 1}{\boldsymbol{\mathscr{R}}}(\boldsymbol{\gamma}, \mathbf{s}) = \underset{n_s \times n_s}{\mathbf{M}}(\boldsymbol{\gamma}, \mathbf{s}, \tau(\mathbf{s})) \underset{n_s \times 1}{\mathbf{s}} - \underset{n_s \times 1}{\mathbf{b}}(\boldsymbol{\gamma}, \mathbf{s}) = \underset{n_s \times 1}{\mathbf{0}} \tag{B.1}$$

$\boldsymbol{\mathscr{R}}$ is the residual vector, $\mathbf{M}$ is the system matrix, $\mathbf{b}$ is the right-hand side forcing vector and $\mathbf{s}$ is the vector of unknowns:

$$\mathbf{s} = \begin{Bmatrix} \mathbf{u} \\ \mathbf{p} \\ \mathbf{t} \end{Bmatrix} \tag{B.2}$$

where $\mathbf{u}$ is the vector of velocity DOFs, $\mathbf{p}$ is the vector of pressure DOFs and $\mathbf{t}$ is the vector of temperature DOFs.

In the most general form; including the Boussinesq coupling, Brinkman term, PSPG and SUPG stabilisation, source terms, as well as design-dependent heat flux loads, the system "stiffness" matrix is given by:

$$\mathbf{M}(\boldsymbol{\gamma}, \mathbf{s}, \tau(\mathbf{s})) = \begin{bmatrix} \mathbf{C}(\mathbf{u}) + \mathbf{K} + \boldsymbol{\alpha}(\boldsymbol{\gamma}) + \mathbf{L}_\delta(\mathbf{u}, \tau(\mathbf{u})) + \mathbf{L}_{\alpha\delta}(\boldsymbol{\gamma}, \mathbf{u}, \tau(\mathbf{u})) & \mathbf{G} + \mathbf{A}_\delta(\mathbf{u}, \tau(\mathbf{u})) & \dots \\ \mathbf{G}^T + \mathbf{L}_\epsilon(\tau(\mathbf{u})) + \mathbf{L}_{\alpha\epsilon}(\boldsymbol{\gamma}, \tau(\mathbf{u})) & \mathbf{A}_\epsilon(\tau(\mathbf{u})) & \dots \\ \mathbf{0} & \mathbf{0} & \dots \\ \dots & \mathbf{B}(\boldsymbol{\gamma}) + \mathbf{Q}_\delta(\boldsymbol{\gamma}, \mathbf{u}, \tau(\mathbf{u})) & \\ \dots & \mathbf{Q}_\epsilon(\boldsymbol{\gamma}, \tau(\mathbf{u})) & \\ \dots & \mathbf{C}_t(\mathbf{u}) + \mathbf{K}_t(\boldsymbol{\gamma}) + \mathbf{Q}_\zeta(\boldsymbol{\gamma}, \mathbf{u}, \tau(\mathbf{u}, \mathbf{t})) & \end{bmatrix} \tag{B.3}$$

and the system right-hand side vector is given by:

$$\mathbf{b}(\boldsymbol{\gamma}, \mathbf{s}, \tau(\mathbf{s})) = \begin{Bmatrix} \mathbf{f} + \mathbf{r}_\delta(\mathbf{u}, \tau(\mathbf{u})) \\ \mathbf{r}_\epsilon(\tau(\mathbf{u})) \\ \mathbf{f}_t(\boldsymbol{\gamma}) + \mathbf{r}_\zeta(\boldsymbol{\gamma}, \mathbf{u}, \tau(\mathbf{u}, \mathbf{t})) \end{Bmatrix} \tag{B.4}$$

where the subscripts of the various stabilisation parameters, τ, have been left out for clarity. As can be seen, the fully coupled system matrix and right-hand side vector are both nonlinear to a large degree. The convection matrices, $\mathbf{C}$ and $\mathbf{C}_t$, are both strongly dependent on the velocity field of the current nonlinear iteration. All of the SUPG stabilisation matrices and vectors, $\square_\delta$, are also dependent on the velocity field, both directly and also implicitly through the calculation of the stabilisation parameter, τ. The PSPG stabilisation matrices and vectors, $\square_\epsilon$, are weakly dependent on the velocity field, implicitly through the stabilisation parameter. The SUPG stabilisation matrix and vector for the temperature field, $\square_\zeta$, is also weakly dependent on the temperature field, implicitly through the stabilisation parameter.

Taking the direct derivative of the residual vector with respect to the state variable vector yields the following:

$$\underset{n_\mathrm{s}\times n_\mathrm{s}}{\frac{d\mathscr{R}}{d\mathbf{s}}} = \underset{n_\mathrm{s}\times n_\mathrm{s}}{\mathbf{M}} + \overbrace{\frac{d\mathbf{M}}{d\mathbf{s}}}^{n_\mathrm{s}\times n_\mathrm{s}}\mathbf{s} - \underset{n_\mathrm{s}\times n_\mathrm{s}}{\frac{d\mathbf{b}}{d\mathbf{s}}} \tag{B.5}$$

where the product rule has been used to calculate the direct derivative of $\mathbf{M}(\boldsymbol{\gamma}, \mathbf{s}, \tau(\mathbf{s}))\,\mathbf{s}$. The vector derivative of the state matrix is a third-order tensor with the size $n_\mathrm{s} \times n_\mathrm{s} \times n_\mathrm{s}$, which when multiplied with the vector of state variables, $\mathbf{s}$, yields an $n_\mathrm{s} \times n_\mathrm{s}$ matrix. The complete product is best illustrated using indicial notation:

$$\frac{d\mathbf{M}}{d\mathbf{s}}\mathbf{s} = \frac{dM_{ik}}{ds_j}s_k \tag{B.6}$$

The direct derivative of the state matrix, $\mathbf{M}(\boldsymbol{\gamma}, \mathbf{s}, \tau(\mathbf{s}))$, is:

$$\frac{d\mathbf{M}}{d\mathbf{s}} = \frac{\partial\mathbf{M}}{\partial\mathbf{s}} + \frac{\partial\mathbf{M}}{\partial\tau}\frac{\partial\tau}{\partial\mathbf{s}} \tag{B.7}$$

and of the state right-hand side vector, $\mathbf{b}(\boldsymbol{\gamma}, \mathbf{s}, \tau(\mathbf{s}))$, is:

$$\frac{d\mathbf{b}}{d\mathbf{s}} = \frac{\partial\mathbf{b}}{\partial\mathbf{s}} + \frac{\partial\mathbf{b}}{\partial\tau}\frac{\partial\tau}{\partial\mathbf{s}} \tag{B.8}$$

Partially differentiating the state matrix, with respect to the state variables, and performing the product in equation (B.6) yields:

$$\frac{\partial\mathbf{M}}{\partial\mathbf{s}}\mathbf{s} = \begin{bmatrix} \frac{\partial\mathbf{C}}{\partial\mathbf{u}}\mathbf{u} + \frac{\partial\mathbf{L}_\delta}{\partial\mathbf{u}}\mathbf{u} + \frac{\partial\mathbf{L}_{\alpha\delta}}{\partial\mathbf{u}}\mathbf{u} + \frac{\partial\mathbf{A}_\delta}{\partial\mathbf{u}}\mathbf{p} + \frac{\partial\mathbf{Q}_\delta}{\partial\mathbf{u}}\mathbf{t} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \frac{\partial\mathbf{C}_t}{\partial\mathbf{u}}\mathbf{t} + \frac{\partial\mathbf{Q}_\zeta}{\partial\mathbf{u}}\mathbf{t} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{B.9}$$

Similarly for the right-hand side:

$$\frac{\partial\mathbf{b}}{\partial\mathbf{s}} = \begin{bmatrix} \frac{\partial\mathbf{r}_\delta}{\partial\mathbf{u}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \frac{\partial\mathbf{r}_\zeta}{\partial\mathbf{u}} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{B.10}$$

Arguing that the scale of magnitude of the stabilisation factors, and thus their derivatives with respect to the velocity and/or temperature fields, are notably smaller than the other contributions to the total derivative, the contribution from the stabilisation factor derivatives are neglected. Finite difference check for the optimisation functionals used throughout this thesis have been accurate with the above neglections, but for pure flow problems, this has been mentioned as an issue in e.g. [38].

Thus, when neglecting the contributions from the derivatives of the stabilisation factors, the partial derivative is equal to the direct derivative and the tangent system matrix is given by:

$$\frac{d\mathscr{R}}{d\mathbf{s}} = \frac{\partial\mathscr{R}}{\partial\mathbf{s}} = \mathbf{M} + \frac{\partial\mathbf{M}}{\partial\mathbf{s}}\mathbf{s} - \frac{\partial\mathbf{b}}{\partial\mathbf{s}} \tag{B.11}$$

where the contributions are given by equations (B.3), (B.9) and (B.10). Due to the locality of the finite element method, the derivative of the complete residual with respect to the full state field vector can be found from the finite element assembly of element residuals differentiated with respect to the element state vector:

$$\frac{d\mathscr{R}}{d\mathbf{s}} = \mathop{\mathsf{A}}_{e=1}^{n_e} \frac{d\mathscr{R}_e}{d\mathbf{s}_e} \tag{B.12}$$

where the element-level residual derivative is given by:

$$\frac{d\mathscr{R}_e}{d\mathbf{s}_e} = \frac{\partial\mathscr{R}_e}{\partial\mathbf{s}_e} = \mathbf{M}^e + \frac{\partial\mathbf{M}^e}{\partial\mathbf{s}_e}\mathbf{s}_e - \frac{\partial\mathbf{b}_e}{\partial\mathbf{s}_e} \tag{B.13}$$

B.1.1 Differentiated element-level matrices

Currently, the contributions to the tangent state matrix from the differentiated state stiffness matrix and right-hand side vector are calculated in MAPLE. The various element matrices and vectors are differentiated by looping over all element state variables and assemblying the matrix contributions by using the product in equation (B.6).

Appendix C

General optimisation theory

C.1 Stationary points

Figure C.1 shows a general one-dimensional nonlinear function exhibiting the various types of stationary points. A stationary point is defined as a point where the gradient of the function is zero and there are three types of stationary points. A minimum extremum, or local minimum, is defined as a point where the gradient of the function changes from negative to positive. A maximum extremum, or local maximum, is defined as a point where the gradient of the function changes from positive to negative. A saddle point, or point of inflection, is defined as a point where the gradient of the function is the same on both sides of the stationary point.

Global maxima and minima are defined as the points within the domain of the function with the maximum or minimum value as compared to the

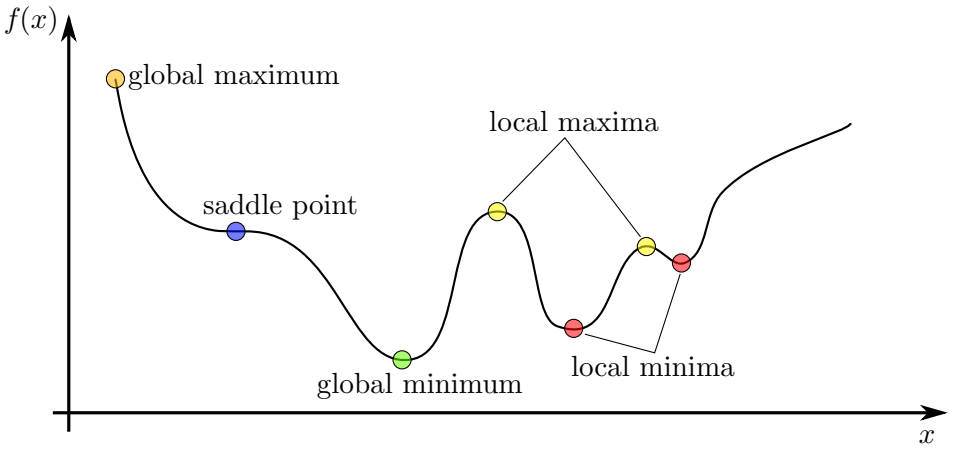


Figure C.1 – A general one-dimensional nonlinear function illustrating the various types of stationary points.

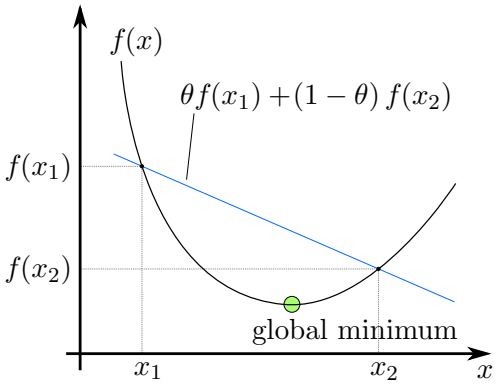


Figure C.2 – A simple one-dimensional convex function.

rest of the domain. The nonlinear function in figure C.1 has a global minimum (green circle) that is also a local minimum. The global maximum (orange circle), however, is not a local maximum. The global maximum and minimum, of a continuous function on a closed interval, are either a local maximum and minimum in the interior of the domain, respectively, or at the end points of the interval.

Figure C.1 clearly illustrates the difficulties posed by nonlinear and non-convex optimisation problems. When solving the constrained topology optimisation problems using gradient-based methods, the converged solution is generally likely to be a local minima and strongly dependent on the initial guess. For instance, if one starts the optimisation algorithm at the right-most edge of the domain, it is very likely that the algorithm gets caught in one of the two local minima. However, if one starts the algorithm at the left-most edge of the domain, it is likely that the algorithm finds its way to the global minimum - if it makes it past the saddle point.

C.2 Convexity

A convex function is defined as a function, $f(x)$, where:

$$\theta f(x_1) + (1-\theta) f(x_2) \geq f(\theta x_1 + (1-\theta) x_2) \tag{C.1}$$

holds for $0 \geq \theta \geq 1$ and all combinations of x_1 and x_2 belonging to the domain of the function, $f(x)$. As illustrated in figure C.3, a graphical interpretation can be seen as a function where a straight line drawn through any two points on the function remains above the function itself and does not cross it inbetween the two points. A very nice, and most important, property of a convex function is that it has a global minimum which is the one and only stationary point, namely a local minimum. It should be noted that a function, $g(x)$, is said to be concave, if the negative of the function, $-g(x)$, is convex.

C.3 Constrained optimisation

A general constrained optimisation problem can be posed as:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimise:}} \quad & f(\mathbf{x}) && \text{(C.2a)} \\ \text{subject to:} \quad & g_i(\mathbf{x}) = 0 \quad \text{for } i = 1, .., m_{eq} && \text{(C.2b)} \\ & h_i(\mathbf{x}) \geq 0 \quad \text{for } i = 1, .., m_{in} && \text{(C.2c)} \end{aligned}$$

where $\mathbf{x}$ is the vector of design variables, f is the objective function, g_i is the m_{eq} number of equality constraints and h_i is the m_{in} number of inequality constraints. The optimisation problem consists of finding the optimal combination of design variables, $\mathbf{x}^*$, that minimises the objective function as well as satisfies the constraints.

The Lagrangian function for the optimisation problem (C.2) is:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \mu_i g_i(\mathbf{x}) - \lambda_i h_i(\mathbf{x}) \tag{C.3}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ are the Lagrange multipliers for the equality and inequality constraints, respectively. The Lagrange function can be seen as a generalised and expanded objective function. The Lagrange function incorporates both information from the original objective function, as well as information about satisfaction of the constraints.

The first-order necessary conditions for constrained optimisation problems are called the Karush-Kuhn-Tucker (KKT) optimality conditions and for the optimisation problem (C.2) they are defined as follows:

$$\begin{aligned} \text{Stationarity:} \quad & \frac{\partial \mathcal{L}}{\partial x_j} = \frac{\partial f}{\partial x_j} - \mu_i \frac{\partial g_i}{\partial x_j} - \lambda_i \frac{\partial h_i}{\partial x_j} = 0 && \text{(C.4a)} \\ \text{Feasibility:} \quad & g_i = 0 && \text{(C.4b)} \\ \text{Feasibility:} \quad & h_i \geq 0 && \text{(C.4c)} \\ & \lambda_i \geq 0 && \text{(C.4d)} \\ \text{Complementarity:} \quad & \lambda_i h_i \geq 0 && \text{(C.4e)} \end{aligned}$$

Thus in constrained optimisation, feasible stationary points of the Lagrange function is sought instead of simply stationary points of the original objective function. Figure C.3 shows a simple two-dimensional inequality constrained optimisation problem. The problem illustrates the fact, that when an optimisation problem becomes constrained, the unconstrained solution often lies outside of the feasible domain for the constrained problem. Instead of satisfying that the gradient of the objective function is zero, the constrained solution satisfies the KKT conditions in equation (C.4). The constrained minimum is thus not a stationary point of the original objective function, but instead a stationary point of the Lagrange function for the optimisation problem.

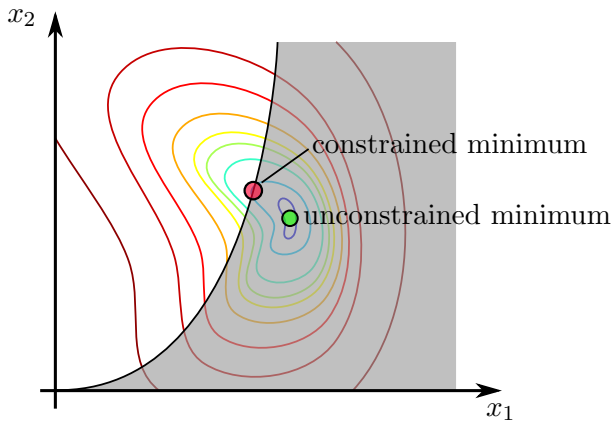


Figure C.3 – Illustration of a simple two-dimensional inequality constrained optimisation problem. The coloured lines are the level contours of the objective function, the black curve is the constraint function and the grayed out part is the infeasible domain.

If the objective function and the negative of the inequality constraints are convex and the equality constraints are affine, then the optimisation problem (C.2) is said to be convex and the optimal solution, $\mathbf{x}^*$, satisfying the KKT-conditions, equation (C.4), are ensured to be the global minimum [42]. This is the essence of which many optimisation algorithms for general nonlinear constrained problems are built on, such as Sequential Quadratic Programming (SQP) algorithms or the Method of Moving Asymptotes (MMA). At each iteration, these algorithms approximate the true nonlinear optimisation problem by a convex approximation. The fact that a convex problem has a single uniquely defined minimum ensures that a solution can be found to the approximation and this solution is used as a good guess for the next iterate of the original nonlinear problem.

There are many ways of solving the optimality conditions in equation (C.4), for instance interior-point and active-set methods, and these are described in any good textbook on the subject, e.g. the book by Nocedal and Wright [42].

Appendix D

Definition of the solid Péclet number

D.1 Non-dimensionalisation

The heat conduction equation for a solid domain is as follows:

$$-k_s \frac{\partial T}{\partial x_i \partial x_i} = s_T \tag{D.1}$$

where k_s is the thermal conductivity of the solid material and s_T is a volumetric heat generation term. The thermal convection-diffusion equation for a fluid domain is given by:

$$\rho_f c_f u_j \frac{\partial T}{\partial x_j} - k_f \frac{\partial T}{\partial x_i \partial x_i} = s_T \tag{D.2}$$

where ρ_f is the density of the fluid, c_f is the specific heat capacity of the fluid, k_f is the thermal conductivity of the fluid and s_T is again a volumetric heat generation term. The similarity between the two governing equations lies in the conduction/diffusion term and the source term. The dissimilarity lies in the convective term.

In order to ensure a smooth continuous transition is possible from the one equation to the other through interpolation, the equations must be non-dimensionalised using the same relations:

$$u_i = U u_i^* \tag{D.3a}$$
$$x_i = L x_i^* \tag{D.3b}$$
$$T = \Delta T\, T^* + T_0 \tag{D.3c}$$
$$s_T = \rho_f c_f \Delta T \frac{U}{L} s_T^* \tag{D.3d}$$

where asterisks (*) denote dimensionless quantities and the volumetric source term has been non-dimensionalised using the fluid density and specific heat

capacity, which will be seen to ensure the same scaling of the effective conductivity for both equations. By inserting the relations in equation (D.3) into equations (D.1) and (D.2) yields:

$$-k_s \frac{\Delta T}{L^2} \frac{\partial T^*}{\partial x_i^* \partial x_i^*} = \rho_f c_f \Delta T \frac{U}{L} s_T^* \tag{D.4a}$$

$$\rho_f c_f U \frac{\Delta T}{L} {u_j}^* \frac{\partial T^*}{\partial x_i^*} - k_f \frac{\Delta T}{L^2} \frac{\partial T^*}{\partial x_i^* \partial x_i^*} = \rho_f c_f \Delta T \frac{U}{L} s_T^* \tag{D.4b}$$

Rearranging the coefficients of the various terms gives rise to the following equations:

$$-\frac{1}{Pe_s} \frac{\partial T}{\partial x_i \partial x_i} = s_T \tag{D.5a}$$

$$u_j \frac{\partial T}{\partial x_j} - \frac{1}{Pe_f} \frac{\partial T}{\partial x_i \partial x_i} = s_T \tag{D.5b}$$

where the solid and fluid Péclet numbers have been defined as:

$$Pe_s = \frac{\rho_f c_f U L}{k_s} \tag{D.6a}$$

$$Pe_f = \frac{\rho_f c_f U L}{k_f} \tag{D.6b}$$

D.1.1 Thermal conductivity ratio

From equation (D.6), it can easily be found that the relation between the solid and fluid Péclet numbers can be expressed as:

$$Pe_s = C_k Pe_f \tag{D.7}$$

where C_k is the ratio between the thermal conductivities of the fluid and solid:

$$C_k = \frac{k_f}{k_s} \tag{D.8}$$

The thermal conductivity ratio for various combinations of metals and fluids are listed in table D.1. Based on the various values, a high contrast value can be seen as $C_k = 10^{-5}$ and a low contrast value as $C_k = 10^{-2}$. The low contrast value will be used in the investigations made for this thesis, unless otherwise stated.

Combination	k_f $[WK^{-1}m^{-1}]$	k_s $[WK^{-1}m^{-1}]$	C_k [dimensionless]
Air - copper	~ 0.025	~ 250	10^{-5}
Air - steel	~ 0.025	~ 40	$6.25 \cdot 10^{-4}$
Air - stainless steel	~ 0.025	~ 16	$1.56 \cdot 10^{-3}$
Air - titanium alloy	~ 0.025	~ 6	$4.17 \cdot 10^{-3}$
Water - copper	~ 0.6	~ 250	$2.4 \cdot 10^{-3}$
Water - steel	~ 0.6	~ 40	$1.5 \cdot 10^{-2}$
Water - stainless steel	~ 0.6	~ 16	$3.75 \cdot 10^{-2}$
Water - titanium alloy	~ 0.6	~ 6	10^{-1}
Engine oil - copper	~ 0.15	~ 250	$6 \cdot 10^{-4}$

Table D.1 – Thermal conductivity ratio for various combinations of metals and fluids. Values estimated from www.engineerstoolbox.com[2], www.wikipedia.com[3] and www.thermalfluidscentral.org[1].

Appendix E

Surface and nodal heat flux

During the derivation of the discrete weak form equations, when integrated by parts, the Laplace operator gives rise to a Neumann boundary term on the heat flux. The dimensional and dimensionless boundary heat flux terms are:

$$b = \int_\Gamma wt\, dS \tag{E.1a}$$

$$b^* = \int_\Gamma wt^*\, dS \tag{E.1b}$$

where t and t^* are the dimensional and dimensionless normal flux along the surface, defined as:

$$t = k_s f_n \tag{E.2a}$$

$$t^* = \frac{1}{Pe_s} f_n^* \tag{E.2b}$$

where it has been assumed that the flux boundary conditions only are applied to the solid domain. f_n and f_n^* are the dimensional and dimensionless temperature gradient normal to the surface:

$$f_n = \frac{\partial T}{\partial x_i} n_i \tag{E.3a}$$

$$f_n^* = \frac{\partial T^*}{\partial x_i^*} n_i \tag{E.3b}$$

Non-dimensionalising the temperature by ΔT and the coordinates by L in equation (E.3a) gives the following relation between the dimensional and dimensionless temperature gradients:

$$f_n = \frac{\Delta T}{L} \frac{\partial T^*}{\partial x_i^*} n_i = \frac{\Delta T}{L} f_n^* \tag{E.4}$$

As the dimensionless normal temperature gradient is what is supplied as the boundary condition in the current implementation, a relation between this

quantity and the dimensional flux is desirable. Combining equations (E.2a) and (E.4) yields:

$$f_n^* = \frac{tL}{k_s \Delta T} \tag{E.5}$$

where t is the dimensional heat flux, L and ΔT is the reference length and temperature difference, respectively, and k_s is the thermal conductivity of the solid. f_n^* can therefore be seen as a Nusselt-number, that is the applied dimensional conductive heat flux scaled by a reference conductive heat flux $\frac{k_s \Delta T}{L}$.

Thus, the temperature scale for problems with only homogenous Dirichlet boundary conditions on the temperature, $T = 0$, and non-homogenous Neumann boundary conditions, $f_n^* \neq 0$, is given by:

$$\Delta T = \frac{tL}{k_s f_n^*} \tag{E.6}$$

Appendix F

Adjoint method

F.1 Brief description of the continuous adjoint approach

To the authors knowledge, the use of the continuous adjoint approach is rather limited in topology optimisation. Othmer [45] introduced the continuous adjoint formulation for the topology optimisation of ducted flows. Deng et al. [23] use the continuous adjoint approach and Taylor-Hood finite elements for topology optimisation of unsteady incompressible flow problems and body-force driven flow problems [24]. Kontoleontos et al. [35] likewise uses the continuous approach, but using the finite volume method.

The continuous adjoint is a viable approach for some implementations, for instance when one does not have access to the underlying discretisation routines. Most proponents of the continuous approach argue that a strength is that another discretisation method can be used for the adjoint problem if a better suited method exists. However, it seems that the adjoint PDE often has so many similarities to the original state PDE, that the discretisation schemes often are the same, or at least very similar. Furthermore, it is argued by Giles and Pierce [30] that the continuous approach is important on a physical level and for an understanding of the underlying properties of the adjoint variables. This insight is gained through the adjoint PDE and its boundary conditions. In the author's opinion, the continuous adjoint is much more cumbersome to derive than its discrete equivalent. To illustrate this, the discrete adjoint problem and sensitivities were derived on two handwritten A4 pages, whereas the continuous problem took ten handwritten A4 pages.

F.2 Lagrange multiplier analogy for adjoint variables

A general discretised PDE-constrained optimisation problem can be written as follows:

$$\text{minimise:} \quad \Phi(\boldsymbol{\gamma}, \mathbf{s}) \tag{F.1a}$$

$$\text{subject to:} \quad \mathscr{R}(\boldsymbol{\gamma}, \mathbf{s}) \tag{F.1b}$$

where $\Phi(\boldsymbol{\gamma}, \mathbf{s})$ is the objective functional, $\boldsymbol{\gamma}$ is the vector of design variables, $\mathbf{s}$ is the vector of state variables and $\mathscr{R}$ is the residual vector for the discretised equation system. The first step of the adjoint method is to form the Lagrangian function for problem (F.1):

$$\underset{scalar}{\mathcal{L}(\boldsymbol{\gamma}, \mathbf{s}, \boldsymbol{\lambda})} = \underset{scalar}{\Phi(\boldsymbol{\gamma}, \mathbf{s})} - \underset{1\times n_\mathrm{s}}{\boldsymbol{\lambda}^\mathrm{T}} \underset{n_\mathrm{s}\times 1}{\mathscr{R}(\boldsymbol{\gamma}, \mathbf{s})} \tag{F.2}$$

where n_s is the number of state DOFs and $\boldsymbol{\lambda}$ is a vector of Lagrange multipliers with entries for each of the equations in the residual vector.

The sensitivities, or gradients with respect to design variables, are found by differentiating the Lagrangian function, equation (F.2), with respect to the design variables:

$$\underset{1\times n_\mathrm{d}}{\frac{d\mathcal{L}}{d\boldsymbol{\gamma}}} = \underset{1\times n_\mathrm{d}}{\frac{d\Phi}{d\boldsymbol{\gamma}}} - \underset{1\times n_\mathrm{s}}{\boldsymbol{\lambda}^\mathrm{T}} \underset{n_\mathrm{s}\times n_\mathrm{d}}{\frac{d\mathscr{R}}{d\boldsymbol{\gamma}}} \tag{F.3}$$

where $\frac{d\square}{d\square}$ denotes the total derivative. The total derivate is defined as:

$$\frac{da}{db} = \frac{\partial a}{\partial b} + \frac{\partial a}{\partial c_i}\frac{\partial c_i}{\partial b} \tag{F.4}$$

where $\frac{\partial\square}{\partial\square}$ denotes the partial derivative and c_i denotes any variables that a explicitly depends on, which are implicit functions of b. Thus, equation (F.3) can be expanded to:

$$\underset{1\times n_\mathrm{d}}{\frac{d\mathcal{L}}{d\boldsymbol{\gamma}}} = \underset{1\times n_\mathrm{d}}{\frac{\partial\Phi}{\partial\boldsymbol{\gamma}}} + \underset{n_\mathrm{s}\times n_\mathrm{d}}{\overset{1\times n_\mathrm{s}}{\frac{\partial\Phi}{\partial\mathbf{s}}}\frac{\partial\mathbf{s}}{\partial\boldsymbol{\gamma}}} - \underset{1\times n_\mathrm{s}}{\boldsymbol{\lambda}^\mathrm{T}}\left(\underset{n_\mathrm{s}\times n_\mathrm{d}}{\frac{\partial\mathscr{R}}{\partial\boldsymbol{\gamma}}} + \underset{n_\mathrm{s}\times n_\mathrm{d}}{\overset{n_\mathrm{s}\times n_\mathrm{s}}{\frac{d\mathscr{R}}{d\mathbf{s}}}\frac{\partial\mathbf{s}}{\partial\boldsymbol{\gamma}}}\right) \tag{F.5}$$

Rearranging equation (F.5) and collecting all the terms containing the derivatives of the state variables yields:

$$\frac{d\mathcal{L}}{d\boldsymbol{\gamma}} = \frac{\partial\Phi}{\partial\boldsymbol{\gamma}} - \boldsymbol{\lambda}^\mathrm{T}\frac{\partial\mathscr{R}}{\partial\boldsymbol{\gamma}} + \left(\frac{\partial\Phi}{\partial\mathbf{s}} - \boldsymbol{\lambda}^\mathrm{T}\frac{d\mathscr{R}}{d\mathbf{s}}\right)\frac{\partial\mathbf{s}}{\partial\boldsymbol{\gamma}} \tag{F.6}$$

Choosing the Lagrange multipliers, $\boldsymbol{\lambda}$, as the solution to the discrete adjoint problem:

$$\frac{d\mathscr{R}}{d\mathbf{s}}^\mathrm{T}\boldsymbol{\lambda} = \frac{\partial\Phi}{\partial\mathbf{s}}^\mathrm{T} \tag{F.7}$$

ensures that the bracket in front of the partial derivatives of the state variables, with respect to the design variables, is equal to zero. This in turn ensures that the gradients of the state variables themselves do not need to be explicitly calculated. Thus, the sensitivities can easily be calculated as:

$$\frac{d\mathcal{L}}{d\boldsymbol{\gamma}} = \frac{\partial \Phi}{\partial \boldsymbol{\gamma}} - \boldsymbol{\lambda}^{\mathrm{T}} \frac{\partial \mathscr{R}}{\partial \boldsymbol{\gamma}} \tag{F.8}$$

which can be seen to be the same as equation (4.19) from section 4.3.2.

Appendix G

Adjoint sensitivities

As described in section 4.3.2, the adjoint sensitivites are found from equation (4.19):

$$\frac{d\mathcal{L}}{d\boldsymbol{\gamma}} = \frac{\partial \Phi}{\partial \boldsymbol{\gamma}} - \boldsymbol{\lambda}^{\mathrm{T}} \frac{\partial \mathscr{R}}{\partial \boldsymbol{\gamma}} \tag{4.19}$$

where $\boldsymbol{\lambda}$ is the solution to the discrete adjoint problem:

$$\frac{d\mathscr{R}}{d\mathbf{s}}^{\mathrm{T}} \boldsymbol{\lambda} = \frac{\partial \Phi}{\partial \mathbf{s}}^{\mathrm{T}} \tag{4.18}$$

The derivatives of the residual with respect to the design variables, $\boldsymbol{\gamma}$, thus needs to be calculated, as well as the derivatives of the various optimisation functionals with respect to both the design variables and the state variables, $\mathbf{s}$. The derivatives are derived in the following sections for the residual and various functionals.

G.1 Derivatives of residual

G.1.1 Residual

The discrete residual vector is as described in section B.1 of appendix B, but for clarity equation (B.1) is repeated here:

$$\underset{n_\mathrm{s}\times 1}{\mathscr{R}(\boldsymbol{\gamma}, \mathbf{s})} = \underset{n_\mathrm{s}\times n_\mathrm{s}}{\mathbf{M}(\boldsymbol{\gamma}, \mathbf{s})} \underset{n_\mathrm{s}\times 1}{\mathbf{s}} - \underset{n_\mathrm{s}\times 1}{\mathbf{b}(\boldsymbol{\gamma}, \mathbf{s})} = \underset{n_\mathrm{s}\times 1}{\mathbf{0}} \tag{B.1}$$

where $\mathscr{R}$ is the residual vector, $\mathbf{M}$ is the system matrix, $\mathbf{b}$ is the right-hand side forcing vector and $\mathbf{s}$ is the vector of unknowns.

G.1.2 Element-level contributions

Due to the locality of both the finite element method and the design parameterisation, the derivative of the complete residual with respect to a single

element design variable, γ_e, receives contributions from only that single element. So the complete residual derivative is found from the finite element assembly of element residual derivatives:

$$\frac{\partial \boldsymbol{\mathscr{R}}}{\partial \boldsymbol{\gamma}} = \mathop{\mathsf{A}}_{e=1}^{n_\mathrm{e}} \frac{\partial \boldsymbol{\mathscr{R}}_e}{\partial \gamma_e} \tag{G.1}$$

where the element-level residual derivative is given by:

$$\frac{\partial \boldsymbol{\mathscr{R}}_e}{\partial \gamma_e} = \frac{\partial \mathbf{M}^e}{\partial \gamma_e}\mathbf{s}_e - \frac{\partial \mathbf{b}_e}{\partial \gamma_e} \tag{G.2}$$

Differentiating the element-level state matrix with respect to the element design variable yields:

$$\frac{\partial \mathbf{M}^e}{\partial \gamma_e} = \begin{bmatrix} \frac{\partial \boldsymbol{\alpha}^e}{\partial \gamma_e} + \frac{\partial \mathbf{L}^e_{\alpha\delta}}{\partial \gamma_e} & \mathbf{0} & \frac{\partial \mathbf{B}^e}{\partial \gamma_e} + \frac{\partial \mathbf{Q}^e_\delta}{\partial \gamma_e} \\ \frac{\partial \mathbf{L}^e_{\alpha\epsilon}}{\partial \gamma_e} & \mathbf{0} & \frac{\partial \mathbf{Q}^e_\epsilon}{\partial \gamma_e} \\ \mathbf{0} & \mathbf{0} & \frac{\partial \mathbf{K}^e_t}{\partial \gamma_e} \end{bmatrix} \tag{G.3}$$

and similarly for the right-hand side:

$$\frac{\partial \mathbf{b}_e}{\partial \gamma_e} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \\ \frac{\partial \mathbf{f}^e_t}{\partial \gamma_e} \end{Bmatrix} \tag{G.4}$$

where the implicit dependence, due to the thermal stabilisation parameter depending on the element $Pe(\gamma_e)$, has been neglected. Thus, the element-level contribution to the derivative of the residual is given by:

$$\frac{\partial \boldsymbol{\mathscr{R}}_e}{\partial \gamma_e} = \begin{Bmatrix} \left(\frac{\partial \boldsymbol{\alpha}^e}{\partial \gamma_e} + \frac{\partial \mathbf{L}^e_{\alpha\delta}}{\partial \gamma_e}\right)\mathbf{u}_e + \left(\frac{\partial \mathbf{B}^e}{\partial \gamma_e} + \frac{\partial \mathbf{Q}^e_\delta}{\partial \gamma_e}\right)\mathbf{t}_e \\ \frac{\partial \mathbf{L}^e_{\alpha\epsilon}}{\partial \gamma_e}\mathbf{u}_e + \frac{\partial \mathbf{Q}^e_\epsilon}{\partial \gamma_e}\mathbf{t}_e \\ \frac{\partial \mathbf{K}^e_t}{\partial \gamma_e}\mathbf{t}_e + \frac{\partial \mathbf{f}^e_t}{\partial \gamma_e} \end{Bmatrix} \tag{G.5}$$

Under simplified conditions, where no source terms are present and the Boussinesq coupling is neglected, the element-level contribution to the derivative of the residual is given by:

$$\frac{\partial \boldsymbol{\mathscr{R}}_e}{\partial \gamma_e} = \begin{Bmatrix} \left(\frac{\partial \boldsymbol{\alpha}^e}{\partial \gamma_e} + \frac{\partial \mathbf{L}^e_{\alpha\delta}}{\partial \gamma_e}\right)\mathbf{u}_e \\ \frac{\partial \mathbf{L}^e_{\alpha\epsilon}}{\partial \gamma_e}\mathbf{u}_e \\ \frac{\partial \mathbf{K}^e_t}{\partial \gamma_e}\mathbf{t}_e \end{Bmatrix} \tag{G.6}$$

Solving the adjoint system for one-way coupled convection-diffusion problems

It is interesting to note, that when using the adjoint method for one-way coupled convection-diffusion problems, where the velocity field drives the

temperature field, the adjoint fields are coupled in the opposite direction. The adjoint equations for one-way coupled convection-diffusion problems become:

$$\frac{\partial \mathscr{R}_t}{\partial \mathbf{t}}^T \boldsymbol{\lambda}_t = \frac{\partial \Phi}{\partial \mathbf{t}}^T \tag{G.7}$$

$$\begin{bmatrix} \frac{\partial \mathscr{R}_u}{\partial \mathbf{u}}^T & \frac{\partial \mathscr{R}_p}{\partial \mathbf{u}}^T \\ \frac{\partial \mathscr{R}_u}{\partial \mathbf{p}}^T & \frac{\partial \mathscr{R}_p}{\partial \mathbf{p}}^T \end{bmatrix} \begin{Bmatrix} \boldsymbol{\lambda}_u \\ \boldsymbol{\lambda}_p \end{Bmatrix} = \begin{Bmatrix} \frac{\partial \Phi}{\partial \mathbf{u}}^T - \frac{\partial \mathscr{R}_t}{\partial \mathbf{u}}^T \boldsymbol{\lambda}_t \\ \frac{\partial \Phi}{\partial \mathbf{p}}^T \end{Bmatrix} \tag{G.8}$$

where it can be seen that the adjoint temperature field gives rise to a forcing for the adjoint velocity field. The above equations are solved in a segregated manor similar to the state equations, but the opposite way around.

G.1.3 Differentiated element-level matrices

Brinkman matrices

$$\underset{24\times 24}{\frac{\partial \boldsymbol{\alpha}^e}{\partial \gamma_e}} = \int_{\Omega^e} \frac{\partial \alpha_e}{\partial \gamma_e} \underset{24\times 3}{\mathbf{N}_u}^{\mathrm{T}} \underset{3\times 24}{\mathbf{N}_u}\, dV \tag{G.9a}$$

$$\underset{24\times 24}{\frac{\partial \mathbf{L}^{\mathbf{e}}_{\alpha\delta}}{\partial \gamma_e}} = \int_{\Omega^e} \frac{\partial \alpha_e}{\partial \gamma_e} \tau_{SU} \underset{24\times 3}{\mathbf{D}_u}^{\mathrm{T}} \underset{3\times 24}{\mathbf{N}_u}\, dV \tag{G.9b}$$

$$\underset{8\times 24}{\frac{\partial \mathbf{L}^{\mathbf{e}}_{\alpha\epsilon}}{\partial \gamma_e}} = -\int_{\Omega^e} \frac{\partial \alpha_e}{\partial \gamma_e} \tau_{PS} \underset{8\times 3}{\mathbf{B}_p}^{\mathrm{T}} \underset{3\times 24}{\mathbf{N}_u}\, dV \tag{G.9c}$$

Conductivity/diffusion matrix

$$\underset{8\times 8}{\frac{\partial \mathbf{K}^e_t}{\partial \gamma_e}} = \int_{\Omega^e} \frac{\partial}{\partial \gamma_e}\left(\frac{1}{Pe_e}\right) \underset{8\times 3}{\mathbf{B}_t}^{\mathrm{T}} \underset{3\times 8}{\mathbf{B}_t}\, dV \tag{G.10}$$

Boussinesq matrices

$$\underset{24\times 8}{\frac{\partial \mathbf{B}^e_t}{\partial \gamma_e}} = \int_{\Omega^e} \frac{\partial Ri_e}{\partial \gamma_e} \underset{24\times 3}{\mathbf{N}_u}^{\mathrm{T}} \underset{3\times 1}{\mathbf{e}_g} \underset{1\times 8}{\mathbf{N}_t}\, dV \tag{G.11a}$$

$$\underset{24\times 8}{\frac{\partial \mathbf{Q}^e_\delta}{\partial \gamma_e}} = \int_{\Omega^e} \frac{\partial Ri_e}{\partial \gamma_e} \tau^e_{SU} \underset{24\times 3}{\mathbf{D}_u}^{\mathrm{T}} \underset{3\times 1}{\mathbf{e}_g} \underset{1\times 8}{\mathbf{N}_t}\, dV \tag{G.11b}$$

$$\underset{8\times 8}{\frac{\partial \mathbf{Q}^e_\epsilon}{\partial \gamma_e}} = \int_{\Omega^e} \frac{\partial Ri_e}{\partial \gamma_e} \tau^e_{PS} \underset{8\times 3}{\mathbf{B}_p}^{\mathrm{T}} \underset{3\times 1}{\mathbf{e}_g} \underset{1\times 8}{\mathbf{N}_t}\, dV \tag{G.11c}$$

For the results shown in this thesis, the Boussinesq forcing is not dependent on the design variables and $\frac{\partial Ri_e}{\partial \gamma_e}$ in the above is thus zero.

G.2 Derivatives of optimisation functionals

G.2.1 Energy dissipation

The total dissipated energy in a domain is defined as follows in continuous form:

$$\Phi = \frac{1}{2}\int_{\Omega}\left(\frac{1}{Re}\frac{\partial u_i}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)+\alpha u_i u_i\right)\,dV \tag{G.12}$$

which when discretised becomes:

$$\Phi(\boldsymbol{\gamma},\mathbf{s}) = \frac{1}{2}\mathbf{u}^{\mathrm{T}}(\mathbf{K}+\boldsymbol{\alpha}(\boldsymbol{\gamma}))\,\mathbf{u} \tag{G.13}$$

where $\mathbf{u}$ is the vector containing the nodal velocities and the matrices, $\mathbf{K}$ and $\boldsymbol{\alpha}$, are the viscosity and Brinkman matrices, respectively, from the finite element discretised equations. Due to the locality of the parameterisation, the total functional can be calculated by the standard summation of the element-level contributions:

$$\Phi(\boldsymbol{\gamma},\mathbf{s}) = \sum_{e=1}^{n_{\mathrm{e}}}\Phi_e(\gamma_e,\mathbf{s}^e) \tag{G.14}$$

where

$$\Phi_e(\boldsymbol{\gamma},\mathbf{s}) = \frac{1}{2}{\mathbf{u}_e}^{\mathrm{T}}(\mathbf{K}^e+\boldsymbol{\alpha}^e(\gamma_e))\,\mathbf{u}_e \tag{G.15}$$

Design derivative of functional

The partial derivative of the dissipated energy functional with respect to a given design variable, γ_e, is easily found to be:

$$\frac{\partial\Phi}{\partial\gamma_e} = \frac{\partial\Phi_e}{\partial\gamma_e} = \frac{1}{2}{\mathbf{u}_e}^{\mathrm{T}}\frac{\partial\boldsymbol{\alpha}^e}{\partial\gamma_e}\mathbf{u}_e \tag{G.16}$$

State derivative of functional: Adjoint RHS

The vector of partial derivatives of the dissipated energy functional with respect to the state variables is found from the finite element assembly of element-level vectors:

$$\frac{\partial\Phi}{\partial\mathbf{s}} = \mathop{\mathsf{A}}_{e=1}^{n_{\mathrm{e}}}\frac{\partial\Phi}{\partial\mathbf{s}_e} \tag{G.17}$$

where the element-level partial derivative vector is given by:

$$\frac{\partial\Phi}{\partial\mathbf{s}_e} = \frac{\partial\Phi_e}{\partial\mathbf{s}_e} = {\mathbf{u}_e}^{\mathrm{T}}(\mathbf{K}^e+\boldsymbol{\alpha}^e(\gamma_e)) \tag{G.18}$$

G.2.2 Maximum allowable fluid volume

The total volume of a given domain is defined as follows in continuous form:

$$V_\Omega = \int_\Omega 1 \, dV \tag{G.19}$$

and the total volume of fluid in the same domain is defined as:

$$V_{\Omega_f} = \int_\Omega \gamma(\mathbf{x}) \, dV \tag{G.20}$$

The functional used to constrain the maximum allowable fluid volume is then defined as:

$$f_{vc} = \frac{V_{\Omega_f}}{\mathrm{v}_f \, V_\Omega} - 1 \tag{G.21}$$

The discretised functional becomes:

$$f_{vc}(\boldsymbol{\gamma}) = \sum_{e=1}^{n_\mathrm{e}} (f_{vc}^e(\gamma_e)) - 1 \tag{G.22}$$

The element-level contribution to the functional is given by:

$$f_{vc}^e(\gamma_e) = \frac{V_e \gamma_e}{V^*} \tag{G.23}$$

where $V^* = \mathrm{v}_f \sum_{e=1}^{n_\mathrm{e}} V_e$ is the prescribed maximum allowable fluid volume and V_e is the discrete element volume found using standard Gauss quadrature.

Design derivative of functional

The partial derivative of the maximum allowable fluid volume functional with respect to a given design variable, γ_e, is easily found to be:

$$\frac{\partial f_{vc}}{\partial \gamma_e} = \frac{\partial f_{vc}^e}{\partial \gamma_e} = \frac{V_e}{V^*} \tag{G.24}$$

State derivative of functional: Adjoint RHS

As the functional does not depend on the state solution, the partial derivative of the functional is zero with respect to all state variables and thus the sensitivities for the volume constraint are constant and only need to be calculated once at the beginning of the optimisation problem.

Solid volume

The procedure is analogous for a constraint on the maximum allowable solid volume, where γ_e in equation G.23 is replaced with $1 - \gamma_e$ and the derivative in equation G.24 thus becomes negative.

G.2.3 Thermal compliance

The thermal compliance for a given problem is defined as follows in continuous form:

$$\Phi = \int_{\Gamma_{flux}} \frac{1}{Pe} \mathbf{N}_t \mathbf{f}_n \, dS \tag{G.25}$$

which when discretised becomes:

$$\Phi(\boldsymbol{\gamma}, \mathbf{s}) = {\mathbf{f}_t}^T \mathbf{t} \tag{G.26}$$

where $\mathbf{t}$ is the vector containing the nodal temperatures and $\mathbf{f}_t$ is the energy flux vector from the finite element discretised equations.

Design derivative of functional

The partial derivative of the thermal compliance is throughout this thesis equal to zero, as the fluxes are all applied to pure solid element and are therefore not design variable dependent through the Péclet number.

State derivative of functional: Adjoint RHS

The vector of partial derivatives of the thermal compliance functional with respect to the state variables is simply:

$$\frac{\partial \Phi}{\partial \mathbf{s}} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{f}_t \end{Bmatrix} \tag{G.27}$$

Appendix H

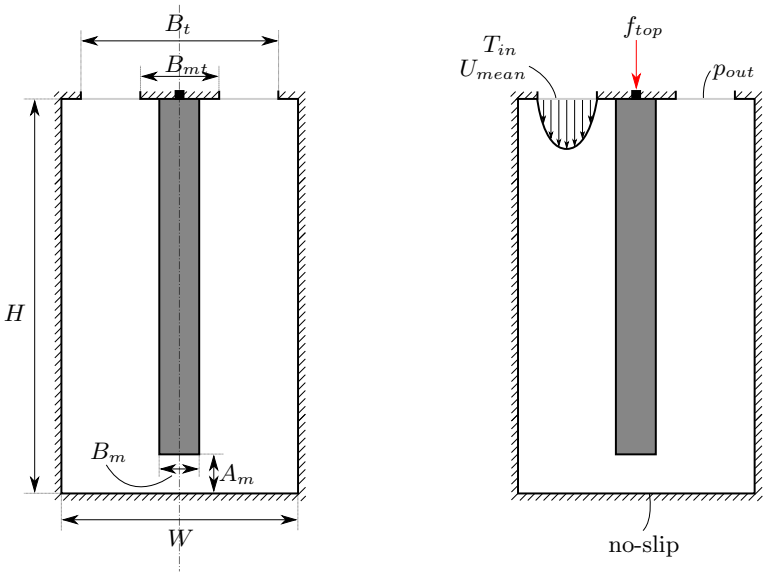
The U-bend problem

H.1 Problem description

The U-bend problem described and used in this chapter is inspired by the "U-bend" example used by Kreissl and Maute [36]. In [36] the example was used to highlight the pressure diffusion that occurs through the porous regions of the design domain when using the Brinkman approach. It should however be noted that the problem is specifically made to highlight the faults of the Brinkman approach. The U-bend problem described in this section will be used to investigate the effects of interpolation functions, maximum Brinkman penalisation parameter and lengthscale.

Figure H.1 shows schematic illustrations of the layout and boundary conditions for the U-bend problem used for the investigations in this chapter. The calculation domain consists of a U-shaped flow domain with an inlet and an outlet, seperated by a porous region in the middle. The flow enters the domain through the left inlet with a parabolic velocity profile with a mean velocity, U_{mean}, and a given inlet temperature, T_{in}, and exits the domain through the right outlet, where a homogenous pressure distribution is prescribed, p_{out}. The rest of the domain boundary is thermally insulated, $f_n = 0$, and has no-slip conditions imposed, $u_i = 0$. A concentrated flux, f_{top}, is imposed on a small solid square at the top-centre of the porous domain. Table H.1 lists the dimensionless quantities specifying the layout and boundary conditions of the U-bend problem. All of the quantities specified are kept constant throughout this investigation, except for the width of the porous region which is varied in order to investigate the effect of lengthscale on the problem. All spatial dimensions are relative to the height of the domain, H, the flow velocities are relative to the mean of the inlet flow, U_{mean}, and the flux and temperature are relative to the scales defined in appendix E. The Reynolds and Péclet numbers are thus defined by the mean inlet velocity and the height of the U-bend.

The problem is particularly nasty because there is only a small throughlet



(a) Sizes **(b)** Boundary conditions

Figure H.1 – Schematic illustration of the sizes and boundary conditions for the U-bend problem.

at the very bottom of the domain and the flow through the porous region needs to be penalised quite aggresively. Furthermore, as a large pressure difference occurs between the inlet and outlet sides, the pressure gradients through the porous region are quite high. Thus, the problem is a difficult, but illustrative, test for the Brinkman penalisation approach.

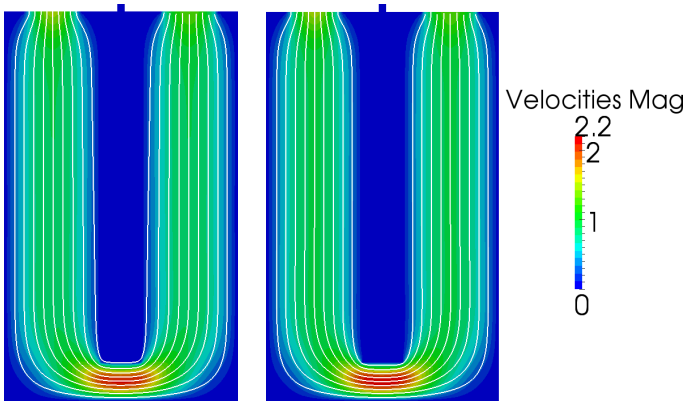
H.1.1 Varying Pe and Re-numbers

The problem is investigated for $Re = Pe = \{1, 10, 100\}$ which is equivalent to increasing the flow velocity while keeping all material properties constant; $Pr = 1$ and $C_k = 10^{-2}$.

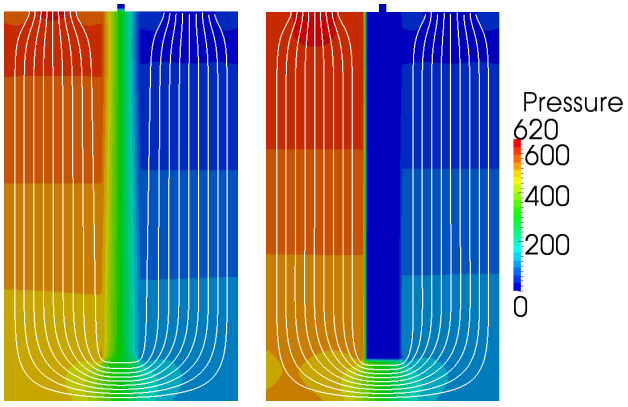
The problem is also investigated for varying $Pe = \left\{1, 10, 10^2, 10^3, 10^4\right\}$ under constant $Re = 1$. In order for this to be equivalent to decreasing the thermal conductivity of the fluid, the thermal conductivity ratio is varied accordingly, $C_k = \left\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\right\}$.

It is important to note that the problem is considered as purely academic at this stage, that is the values of the various parameters are not chosen to represent any specified physical situation. But the values are in the range of realistic problems and hence representative of the problems that can be physically encountered.

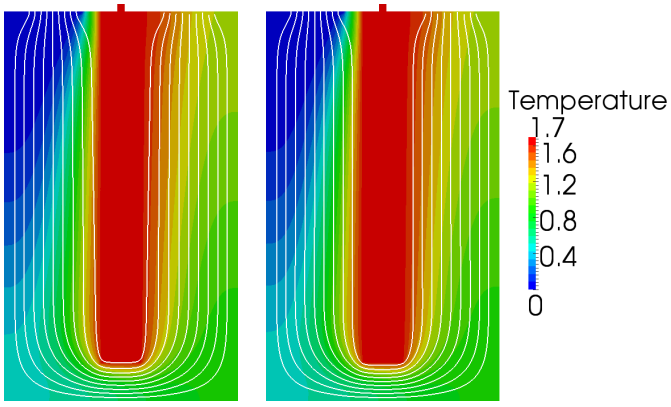
Figure H.2 shows the state fields for the U-bend problem at $Re = 1$ and $Pe = 100$, using both the Brinkman penalisation and a body-fitted mesh.



(a) $\overline{\alpha} = 10^6$ **(b)** Body-fitted



(c) $\overline{\alpha} = 10^6$ **(d)** Body-fitted



(e) $\overline{\alpha} = 10^6$ **(f)** Body-fitted

Figure H.2 – Velocity, pressure and temperature fields for the Brinkman-penalised and body-fitted meshed at $Re = 1$ and $Pe = 100$.

Sizes	
Height:	$H = 1$
Width:	$W = 0.6$
Top:	$B_t = 0.5$
	$B_{mt} = 0.2$
Porous width:	$B_m = 0.02 - 0.18$
Throughlet height:	$A_m = 0.1$

(a) Sizes

Boundary conditions	
Inlet:	$U_{max} = 1.5$
	$T_{in} = 0$
Outlet:	$p_{out} = 0$
Flux:	$f_{top} = 10^{-2}$

(b) Boundary conditions

Table H.1 – The dimensionless quantities used for the U-bend problem shown in figure H.1.

It can be seen that all of the fields look very similar as they should, the only significant differences are in the maximum velocity at the throughlet and in the pressure fields. It can be seen that for the Brinkman approach, a continuous pressure field exists through the porous bar and that the pressure field is a little lower than for the body-fitted mesh because of this. This is the pressure diffusion that Kreissl and Maute [36] mention. It should be noted that even though it appears that there is a zero-valued velocity and pressure solution inside the solid bar of the body-fitted mesh, this is only because ParaView visualised the non-existant solution as zero.

H.2 Effectiveness of Brinkman penalisation

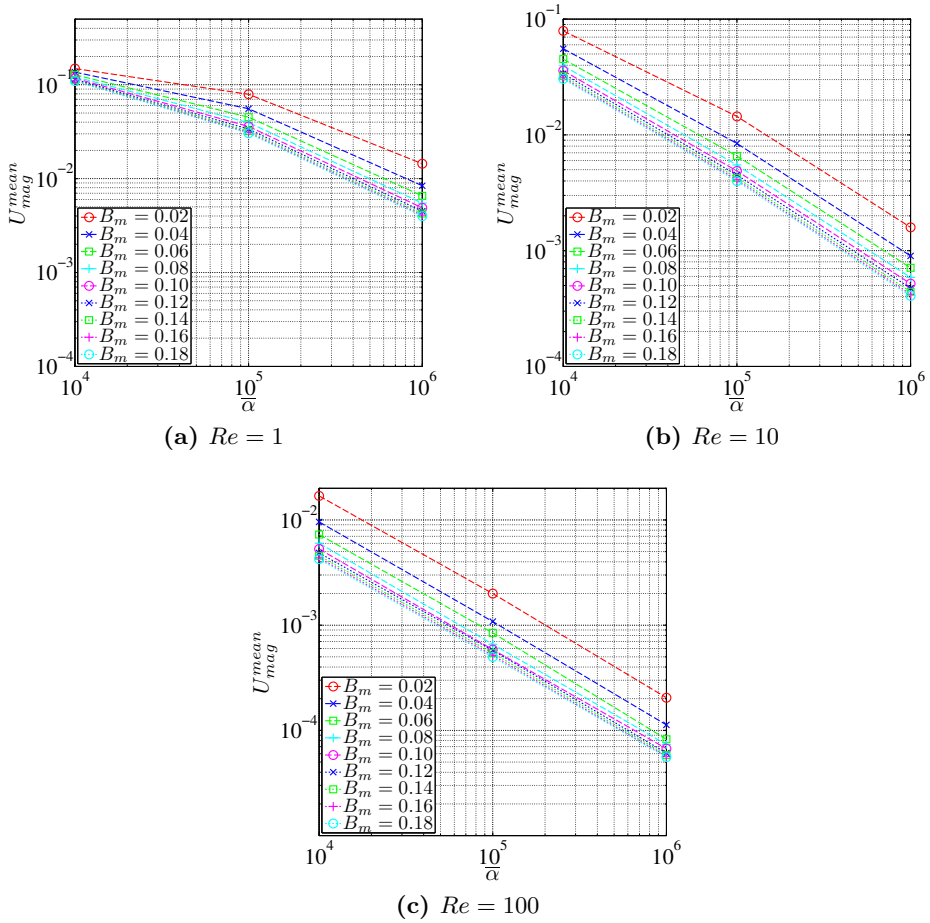
In this section, the effectiveness of the Brinkman penalisation, for driving velocities towards zero in the solid regions of the domain, is investigated.

H.2.1 Velocity magnitude in porous region

Variable maximum penalisation

The maximum penalisation factor for the Brinkman penalisation, $\overline{\alpha}$, is very important for topology optimisation purposes. As the main purpose of the Brinkman penalisation is to drive velocities towards zero in the solid regions of the design domain, the maximum penalisation factor needs to be carefully chosen. It has to be large enough to ensure small velocities in the solid regions, while not being too large so as to degrade the condition of the state matrix and thus have a negative effect on the convergence of the nonlinear Newton solver.

The U-bend problem is analysed for three values of $\overline{\alpha}$ and for three different Reynolds numbers, $Re = 1, 10, 100$. In order to compare the different runs, the mean of the velocity magnitude, for all nodes belonging to



(a) $Re = 1$ **(b)** $Re = 10$

(c) $Re = 100$

Figure H.3 – The node-wise mean of the velocity magnitude in the porous region as a function of the maximum penalisation factor, shown for several values of B_m.

elements in the porous region, is found. Figure H.3 shows the mean velocity magnitude as a function of the $\overline{\alpha}$ for various Re and B_m. It can be seen that the velocity magnitude is pushed towards zero as $\overline{\alpha}$ is increased, which is exactly as expected. Furthermore, it can be seen that the velocity magnitude decreases as the Re-number is increased. This is emphasised by figure H.4 which shows that the mean velocity magnitude decreases very similarly as a function of Re-number and $\overline{\alpha}$. Although the two figures look identical, there are slight differences in the error values. The fact that the two graphs are almost identical shows that the penalisation factor needed to obtain a certain mean velocity magnitude is proportional to inverse of the Re-number, as suggested by the physical scaling of the penalisation parameter, as described in section 2.5. This means, for example, that the same mean velocity magnitude is achieved for $Re = 10$ using $\overline{\alpha} = 10^6$ as for

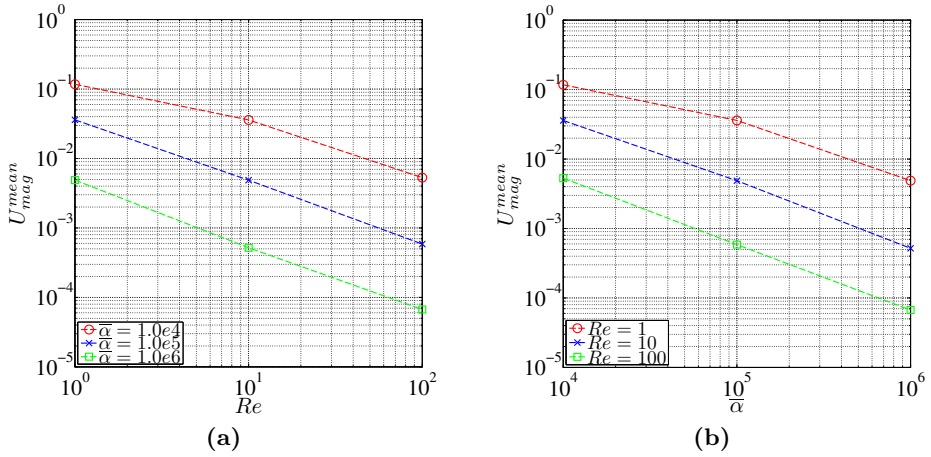


Figure H.4 – The node-wise mean of the velocity magnitude in the porous region as a function of **(a)** the Reynolds number and **(b)** the maximum penalisation factor, shown for $B_m = 0.1$.

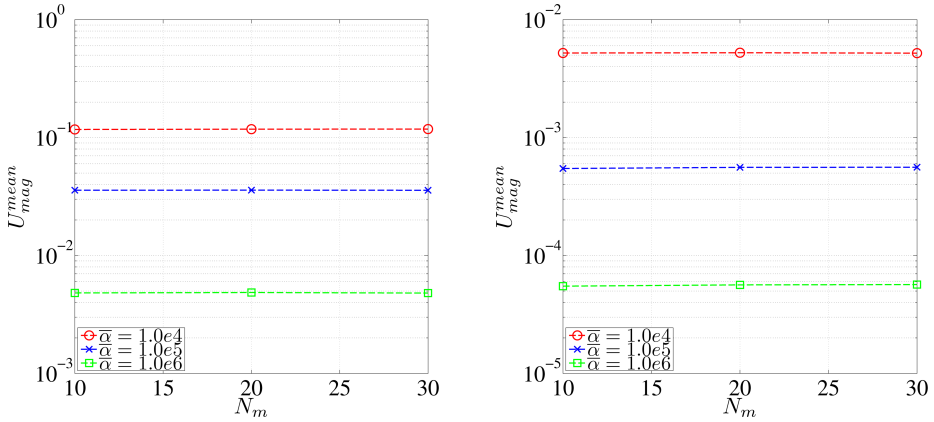
$Re = 100$ using $\overline{\alpha} = 10^5$.

As noted in section 2.5, physically it makes sense that a higher penalisation factor is needed for low Reynolds number problems, as the flow of a fluid in a porous medium is diffusion-dominated. Thus, when the problem becomes convection-dominated, the diffusion of flow into the porous material decreases.

It is important to note, that the maximum penalisation coefficient must not be set too high. A high penalisation coefficient degrades the convergence of the nonlinear solver and more importantly the derivatives of the porosity with respect to design variable increases with respect to the maximum penalisation coefficient. Therefore, only the three penalisation coefficients, $\overline{\alpha} = \left\{10^4, 10^5, 10^6\right\}$, and no higher are investigated.

Variable mesh size

As figure H.5 shows, the mesh size does not really affect the mean velocity magnitude in the porous region. However, figure H.6 shows that oscillations occur at the lower and upper end of the porous region, $x_2 \approx 0.1$ and $x_2 \approx 1.0$, and that these are smoothed out when using a finer mesh. This makes sense as the oscillations occur due to large velocity gradients forming perpendicular to the flow directions along the fluid-solid interfaces. Seo [48] and Lee [38] note and investigate oscillations in the state fields due to the large gradients in porosity, if too large a maximum Brinkman penalisation coefficient is used. These large gradients in material, going from 0 to 1 across an interface, lead to large gradients in the velocity solution and thus oscillations, if the mesh is not sufficiently refined. In [48] the oscillations



(a) $Re = 1$ **(b)** $Re = 100$

Figure H.5 – The node-wise mean of the velocity magnitude in the porous region as a function of number of elements across the porous region, shown for $B_m = 0.1$.

are attempted dampened by interpolating the viscosity and in [38] the oscillations in the state field are not investigated, but the subsequent design oscillations are avoided by introducing a move-limit for the design variables. The problem is also addressed by Kreissl et al. [37], where the oscillations are blamed on the SUPG stabilisation and this is therefore removed for the elements with large Brinkman penalisation coefficients.

Initially significant convergence issues were experienced for the problems shown in this appendix. The oscillations at the solid-fluid interfaces travelled into the porous region and were not as localised as seen in figure H.6. Furthermore, the pressure distribution was also oscillatory and exhibited large variations at the solid-fluid interfaces. Both of these issues were mitigated by making the stabilisation parameter, τ_{PS}, dependent on the porosity, as explained in appendix A.7.

There is next to no doubt that the oscillations that remain are occuring due to the sharp gradients in material distribution, and thus the velocity field, and that mesh refinement would help on the issue. However, for coarse meshes, these oscillations and design convergence problems might be addressed by using additional specialised stabilisation techniques for the purpose. The author has tested various discontinuity-capturing stabilisation methods, such as DCDD- or (Y, Z, β)-stabilisation [60, 61], that add stabilisation in the discontinuity directions, but with no success. This is likely because the velocities inside of the Brinkman solid are much smaller than in the fluid domain and this, coupled with the fact that the ordinary stabilisation terms are many orders of magnitude lower the Brinkman penalisation terms, basically leads to the discontinuity-capturing methods having no effect inside of the Brinkman solid.

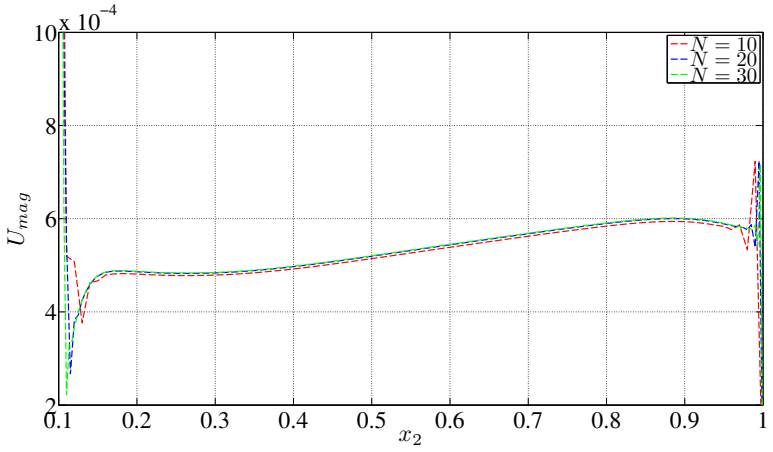
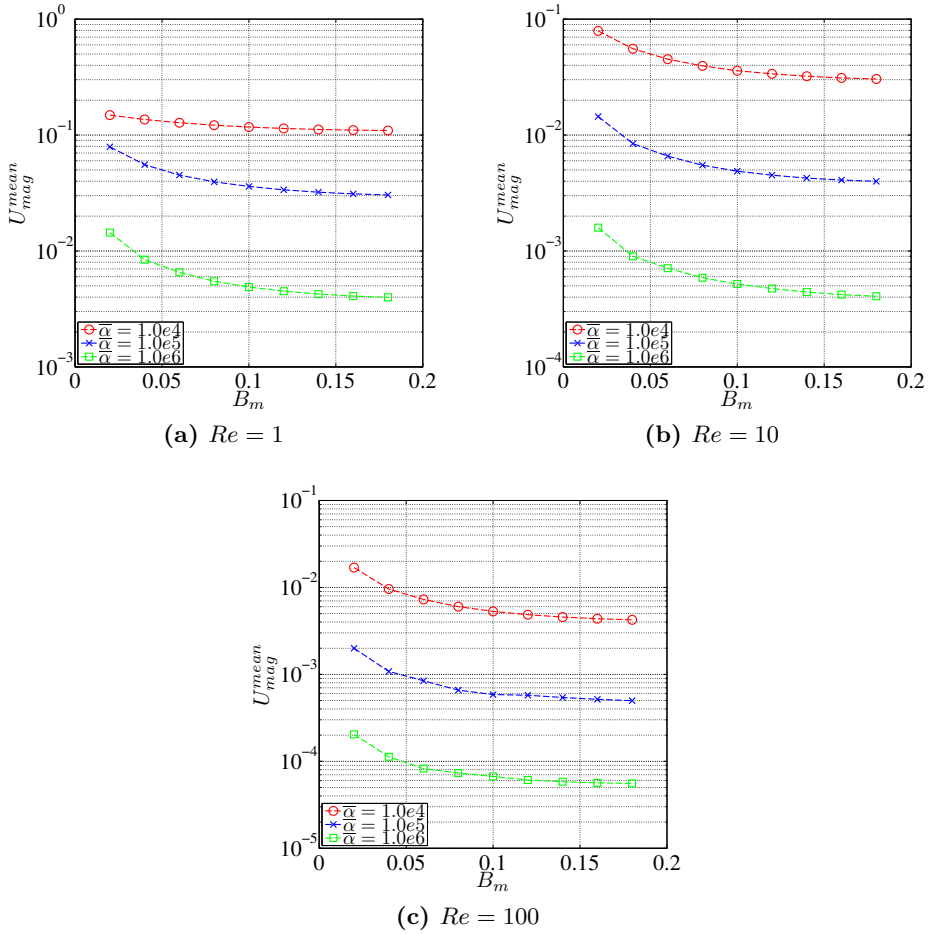


Figure H.6 – The velocity magnitude across the vertical midline for the porous region as a function of vertical coordinate, x_2, shown for $Re = 100$, $\overline{\alpha} = 10^5$, $B_m = 0.1$ and various meshes.

Variable width of porous region

It is noted in [36] that when representing thin members, that ought to inhibit the flow, using the Brinkman approach, one obtains inaccurate and unphysical solutions due to the penalisation not effectively modelling an impenetrable solid. Therefore, it is thought to be interesting to investigate the effect of lengthscale on the velocity magnitude in the porous region.

As can already be seen from figure H.3, it appears that increasing the width of the porous region, while maintaining the same $\overline{\alpha}$, decreases the mean velocity magnitude in the porous region. Figure H.7 shows the mean velocity magnitude as a function of the porous region width for various Re and $\overline{\alpha}$. The general trend can be seen to be that the velocity magnitude is generally reduced by a magnitude of about $\sim 5 - 7 \cdot 10^{-1}$ by increasing the width from 0.02 to 0.18, when the Brinkman penalisation is working efficiently; that is neglecting $Re = 1$ with $\overline{\alpha} = 10^4$ where the penalisation is not working and there is active flow through the porous region. This suggests that it would be beneficial to impose a lengthscale during topology optimisation in order to ensure that small members, that do not effectively inhibit the flow, from being formed. It would be optimal to impose as large a lengthscale as possible, but the larger a lengthscale being imposed, the lesser design freedom for the optimisation process. It can also be seen that the largest reduction in magnitude occurs from inhibiting the smallest lengthscales, $B_m = 0.02 - 0.08$.



(a) $Re = 1$ **(b)** $Re = 10$

(c) $Re = 100$

Figure H.7 – The node-wise mean of the velocity magnitude in the porous region as a function of the width of the porous region, shown for various Re-numbers.

H.2.2 Accuracy of solutions

In order to investigate the effectiveness of the Brinkman penalisation to force the solutions towards those of a body-fitted mesh, comparisons are made for the various subproblems. Body-fitted denotes a computational mesh where the porous region is modelled as a true solid material, that has neither velocity nor pressure DOFs, but only has temperature DOFs. For the comparisons, only the flow solutions, velocity and pressure, in the flow domain are considered. That is, the velocities and pressure in the porous region are not compared to the body-fitted values as they are zero/non-existant. The temperature solution for the entire domain is used in the comparisons.

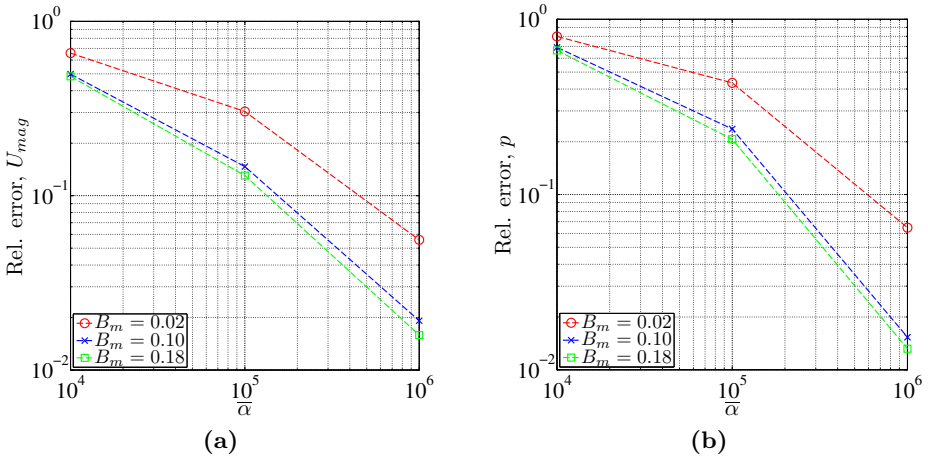


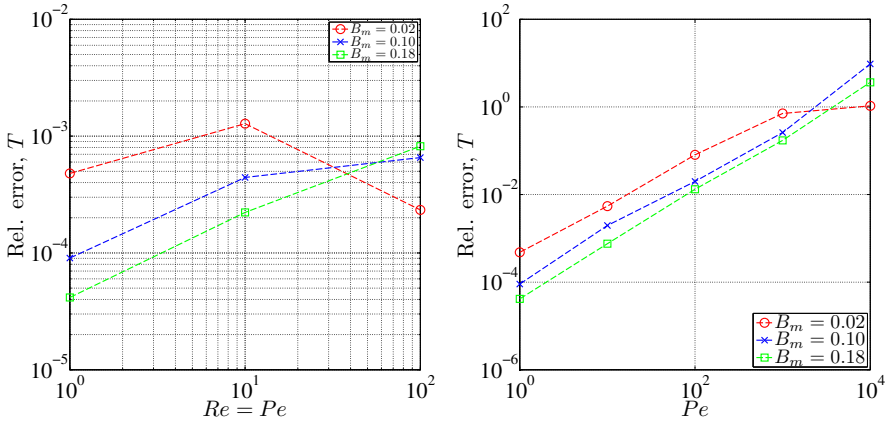
Figure H.8 – The maximum relative error, between the Brinkman penalised flow solution and a body-fitted equivalent, as a function of $\overline{\alpha}$. Shown for several B_m and $Re = Pe = 1$.

Variable maximum penalisation

Figure H.8 shows as expected, that the accuracy of the flow solution, the velocity and pressure fields, increases as the maximum penalisation coefficient is increased. The figure shows the results for $Re = 1$ but is representative for the larger values too. But, it is again observed that keeping $\overline{\alpha}$ constant, the accuracy improves as the Re-number is increased, due to the Brinkman penalisation becoming more effective for larger Re-numbers.

Variable Pe-number

Figure H.9 shows the maximum relative error between the temperature fields for the Brinkman penalised solutions and body-fitted equivalents. The figure only shows the results for $\overline{\alpha} = 10^6$, but they are representative of the general trend. It can be seen that generally, increasing the Pe-number leads to an increase in the relative error between the penalised solutions and the body-fitted reference solutions. The effect is seen to be very pronounced for the case of variable Pe under constant Re, as can be seen in figure H.9b. It makes sense that the temperature solution becomes more inaccurate as the Pe-number is increased, as the heat transfer becomes convection-dominated. This means that even small, otherwise neglectable, velocities in the porous regions lead to unphysical convective heat transfer in these regions. It is important to note that the case of variable Pe under constant Re is especially nasty, because when $Re = 1$ the velocities in the porous region are quite high and the beforementioned effect is highlighted when increasing the Pe-number. To illustrate that these extreme cases used are physically realistic,



(a) Variable $Re = Pe$ **(b)** Constant Re, variable Pe

Figure H.9 – The maximum relative error, between the Brinkman penalised temperature solution and a body-fitted equivalent, as a function of Pe-number. Shown for several B_m and $\overline{\alpha} = 10^6$.

unused engine oil has a Prandtl number ranging from $Pr \approx 1 \cdot 10^2 - 1 \cdot 10^5$ [1] for various temperatures and coupled with copper yields $C_k \approx 6 \cdot 10^{-4}$, appendix D, which both lie in the region of the shown examples. Thus, for flows with a low Re-number and a high Pr-number, it might be neccessary to penalise the convective term of the convection-diffusion equation further.

H.3 Effects of interpolation

The goal of topology optimisation is to end up with binary designs, where all design variables either take the value 0 for solid or 1 for fluid. Thus, it is important that the physical modelling is correct for these two extremes in order for the final optimised design to be physically realistic.

However, when performing topology optimisation using continuous design variables, as is often the case, the interpolation between the two extremes is also of utmost importance. It can be discussed whether the intermediate regions should be physically realistic or not, but when the end goal is to have binary designs, the most important thing is to make sure that the intermediate regions are unattractive with respect to the optimisation problem.

In this section, three prospective objective functionals for topology optimisation of fluid-solid thermal problems are considered. The variation of each functional with respect to a single design variable is investigated. The U-bend problem, as described in section H.1, is used, but instead of the porous region having constant maximum Brinkman penalisation coefficient and solid Péclet number, the porosity and Péclet number are interpolated using the functions described in sections 4.2.1 and 4.2.2. A single design variable

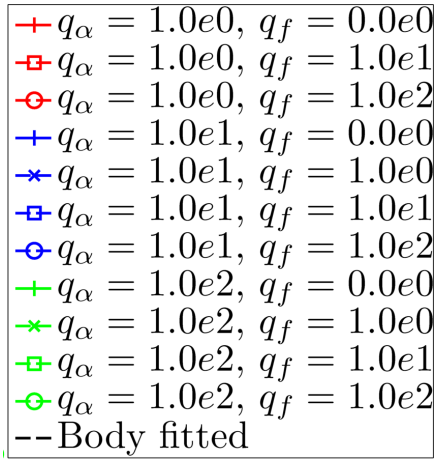
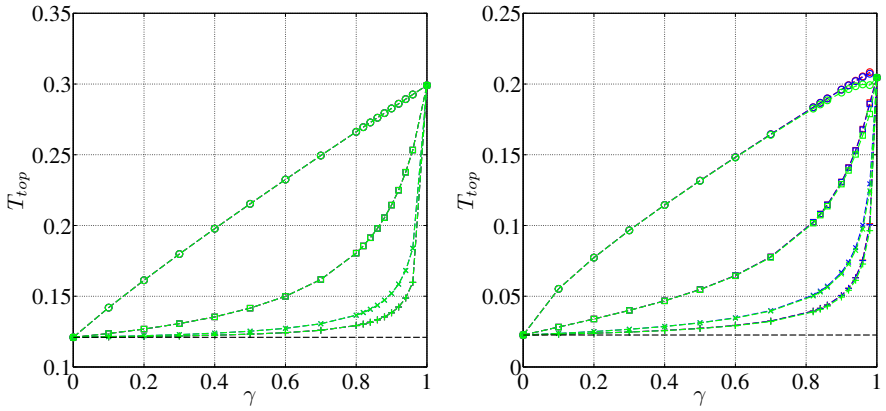


Figure H.10 – Legend for the variable design variable plots showing the different combinations of interpolation penalisation factors used. The datasets for $q_\alpha = q_f = 1 \cdot 10^0$ are not shown due to unforeseen problems with the saved datasets and time was not available to restore them.

is used for the entire porous region and this is varied between 0 and 1. The design variable sweep is carried out for the interpolation penalisation factor combinations shown in figure H.10 with constant $\overline{\alpha} = 10^6$ and for $B_m = 0.1$. The common legend for all figures is shown a single time in figure H.10 in order to save space on the following pages. It is important to note that in this section, the functionals are considered as univariate, so the properties shown here may or may not be carried over for use for multivariate topology optimisation. It will be assumed that the behaviour will be somewhat similar.



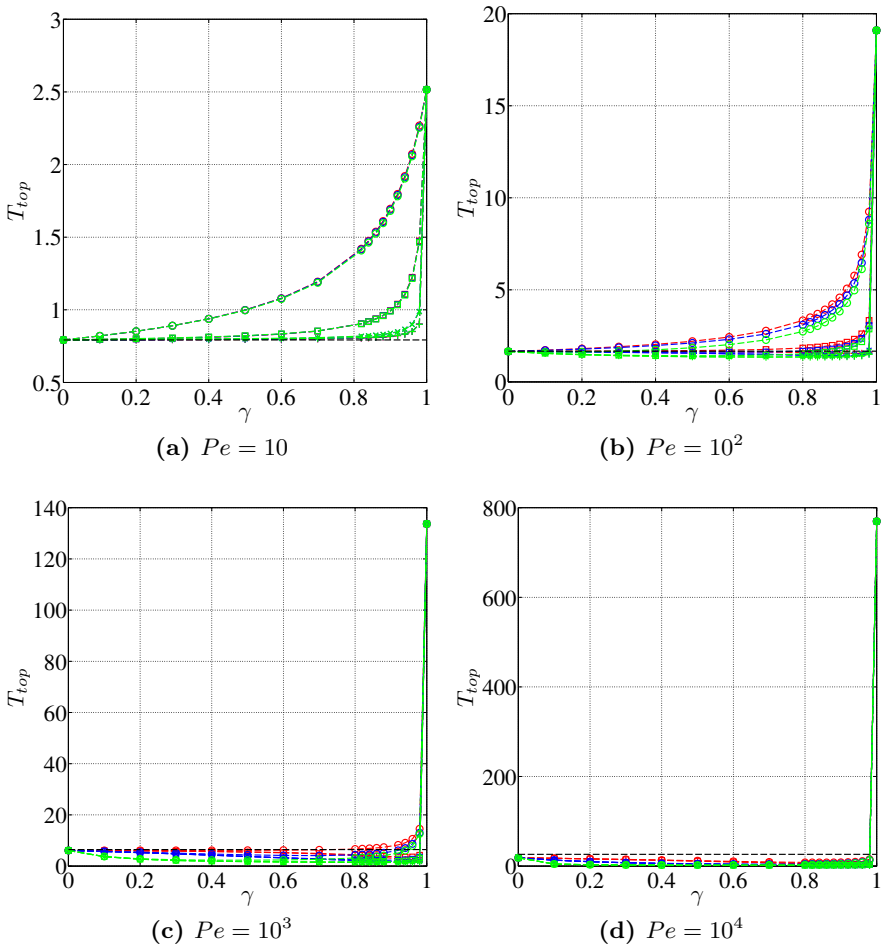
(a) $Re = Pe = 1$ **(b)** $Re = Pe = 10^2$

Figure H.11 – Temperature at flux, T_{top}, as a function of design variable for varying Re- and Pe-numbers together. Only shown for $Re = Pe = \{1, 10^2\}$ due to the graph for $Re = Pe = 10$ being more or less identical.

H.3.1 Temperature at flux, T_{top}

The first objective function is the temperature at the point of the concentrated flux. This is seen as an obvious candidate for optimisation of fluid-solid thermal problems under flux loading, where the temperature of the solid region is of particular interest. This function has been succesfully used as the objective functional in the topology optimisation of pure conductive heat transfer problems, e.g. [13] and by the author in an earlier project [5]. For a single constant concentrated flux, this function is equivalent to the thermal compliance.

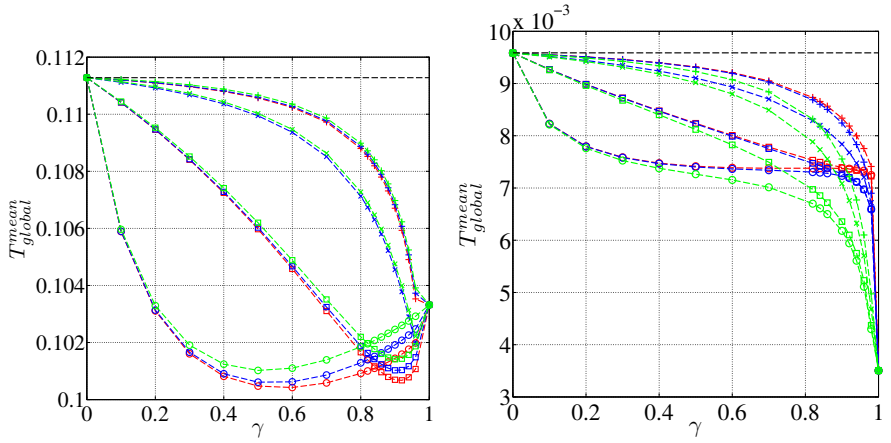
Figure H.11 shows the variation of the function for varying the Re- and Pe-numbers together and figure H.12 for varying Pe-number under constant $Re = 1$. It can be seen that the function is generally well-behaved and monotonously increasing for all cases, except for the high Pe-cases where $Re = 1$. The nice properties should, however, be regained by using a high penalisation factor for the interpolation of the effective conductivity, q_f. The reason for the increasingly sharp gradient at $\gamma = 1$ is very likely due to the fact that C_k is changing orders of magnitude when the conductivity of the fluid is changed and q_f should be increased respectively to achieve the same penalisation of the effective conductivity. Time has unfortunately not allowed these dataset to be rerun after this discovery. Thus, minimising the temperature at the concentrated flux appears to be a well-posed problem that is likely to end up with binary designs for a large enough penalisation factor.



(a) $Pe = 10$ **(b)** $Pe = 10^2$

(c) $Pe = 10^3$ **(d)** $Pe = 10^4$

Figure H.12 – Temperature at flux, T_{top}, as a function of design variable for varying Pe-number under constant $Re = 1$.



(a) $Re = Pe = 1$ **(b)** $Re = Pe = 10^2$

Figure H.13 – Global mean temperature, T^{mean}_{glob}, as a function of design variable for varying Re- and Pe-numbers together. Only shown for $Re = Pe = \{1, 10^2\}$ due to the graph for $Re = Pe = 10$ being similar to $Re = Pe = 1$.

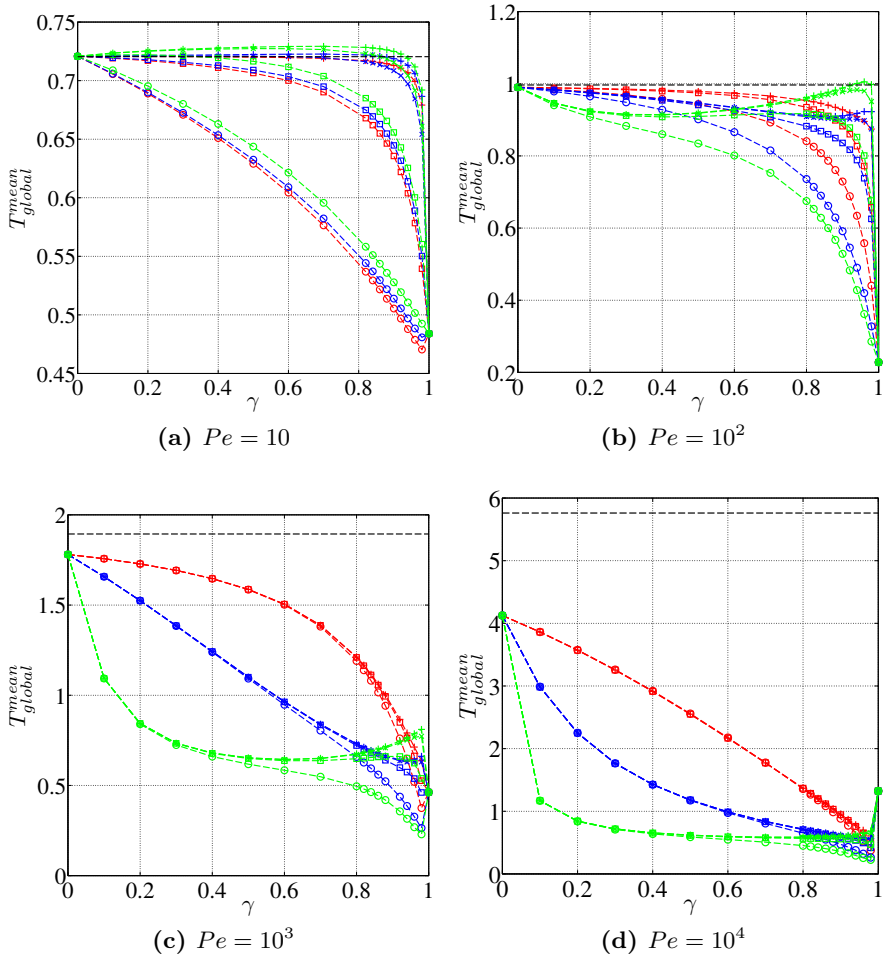
H.3.2 Global volumetric mean temperature, T^{mean}_{glob}

The volume-averaged temperature of the entire domain, from now on termed the global mean temperature, is considered as an objective functional based on its use in Lee [38]. In [38] the conductivity is interpolated using the RAMP function, as in this thesis, and interesting results for the design of convective cooling systems are presented. However, several problems are encountered, such as a lot of intermediate design variables causing unphysical, but favourable, flow conditions. The functional in itself seems reasonable, as minimising the mean temperature can be seen as a combination of lowering the solid temperature and raising the fluid temperature for solid thermal problems under flux loading and fluid cooling, as considered in [38].

Figure H.13 shows the variation of the functional for varying the Re- and Pe-numbers together and figure H.14 for varying Pe-number under constant $Re = 1$. It can be seen that the functional generally behaves very different to the temperature at the flux, T_{top}.

It is interesting to see that the functional goes from being weakly dependent on the porosity penalisation factor, q_α, and strongly dependent on the penalisation of the effective conductivity, q_f, for low Pe-numbers, e.g. figure H.13a, to being strongly dependent on q_α and weakly dependent on q_f, e.g. figures H.14c and H.14d for high Pe-numbers. This makes sense as the temperature field becomes convection-dominated and thus more dependent on the velocity field, which of course is tightly coupled to the interpolation of the porosity and thus q_α.

Generally, it can be seen that the functional is not very well suited for

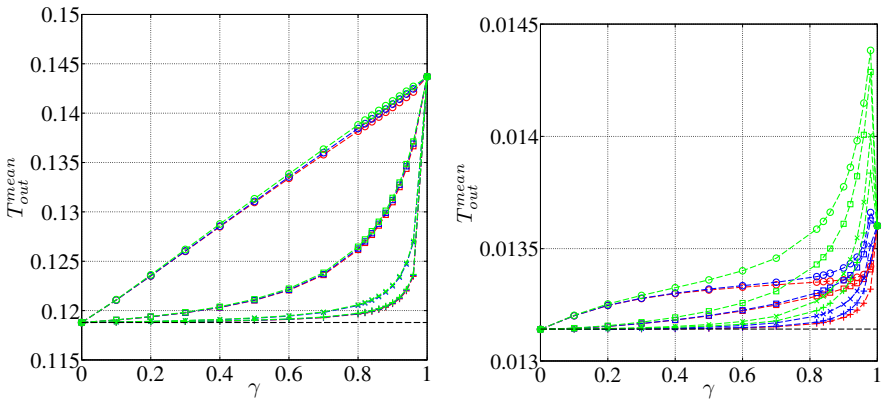


(a) $Pe = 10$ **(b)** $Pe = 10^2$

(c) $Pe = 10^3$ **(d)** $Pe = 10^4$

Figure H.14 – Global mean temperature, T^{mean}_{glob}, as a function of design variable for varying Pe-number under constant $Re = 1$.

minimisation problems without taking careful consideration of the interpolation penalisation parameters. The functional is non-monotonously decreasing for a wide selection of the problems and parameter combinations and this shows that the functional needs to be used with care. In [38], the functional is used for different problems, for which most of the results show excessive amounts of intermediate design variables and thus unphysical flow conditions. This is mitigated by providing a non-designable flow path through the domain and also by including a weighting of the dissipated energy of the flow in the objective functional. It is postulated that the problems encountered in [38] could be due to global mean temperature being ill-posed as an objective functional for many combinations of interpolation penalisation parameters. The function seems to be exhibiting extremely rapid variations near $\gamma = 1$ which need to be further investigated before this function can be considered for use, it could be due to too rapid variation in the order of magnitude of the effective conductivity and thus a larger penalisation factor, q_f, is needed.



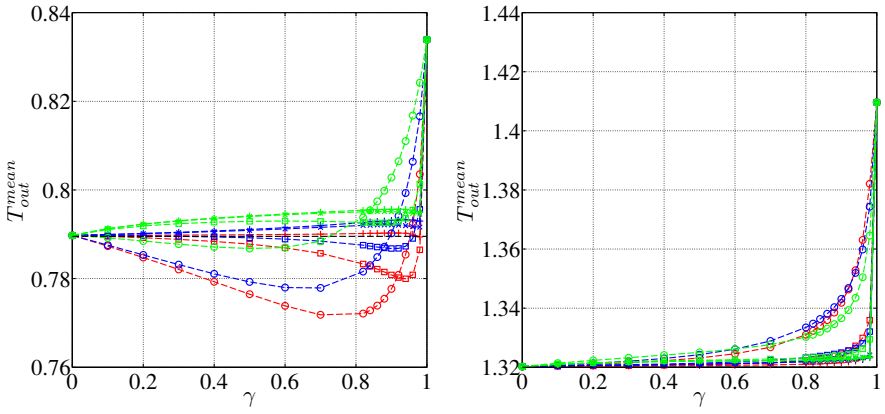
(a) $Re = Pe = 1$ **(b)** $Re = Pe = 10^2$

Figure H.15 – Mean outlet temperature, T_{out}^{mean}, as a function of design variable for varying Re- and Pe-numbers together. Only shown for $Re = Pe = \{1, 10\}$ due to the graph for $Re = Pe = 10^2$ being similar to $Re = 1, Pe = 10$, figure H.16a.

H.3.3 Mean outlet temperature, T_{out}^{mean}

The mean temperature of the fluid at the outlet is the third and final objective functional considered as part of this investigation. It is seen as a possible candidate for the objective functional for a maximisation problem, for applications where a fluid is to be heated up by passing through channels in a heated solid device.

Figure H.15 shows the variation of the functional for varying the Re- and Pe-numbers together and figure H.14 for varying Pe-number under constant $Re = 1$. It can be seen that the functional generally behaves very poorly. It exhibits non-monotonously increasing behaviour for most problems and parameter combinations. Thus, it would appear that the mean outlet temperature is not very well-suited as an objective functional for a maximisation problem. However, it should be noted that for $Re = Pe = 1$ and $Re = Pe = 10^2$, figures H.15a and H.15b respectively, the behaviour is quite similiar to that of T_{top}.



(a) $Pe = 10$ **(b)** $Pe = 10^2$

(c) $Pe = 10^3$ **(d)** $Pe = 10^4$

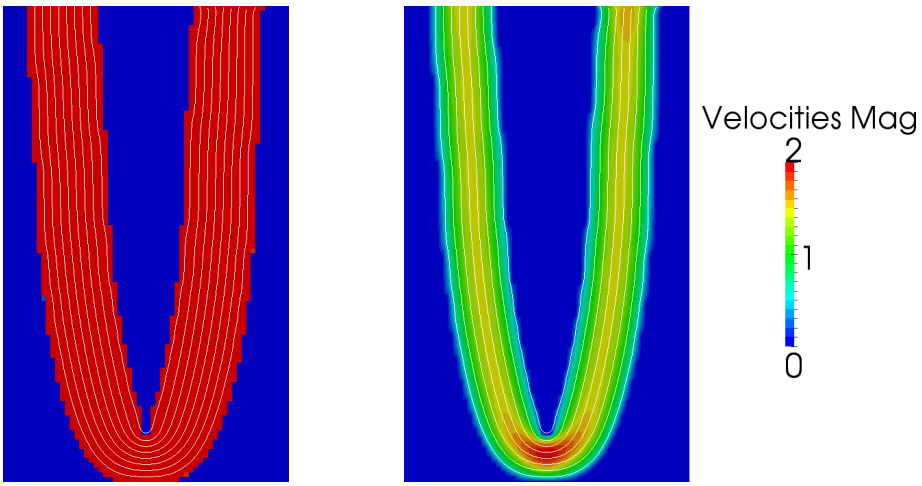
Figure H.16 – Mean outlet temperature, T_{out}^{mean}, as a function of design variable for varying Pe-number under constant $Re = 1$.

H.4 U-bend: Minimum dissipated energy

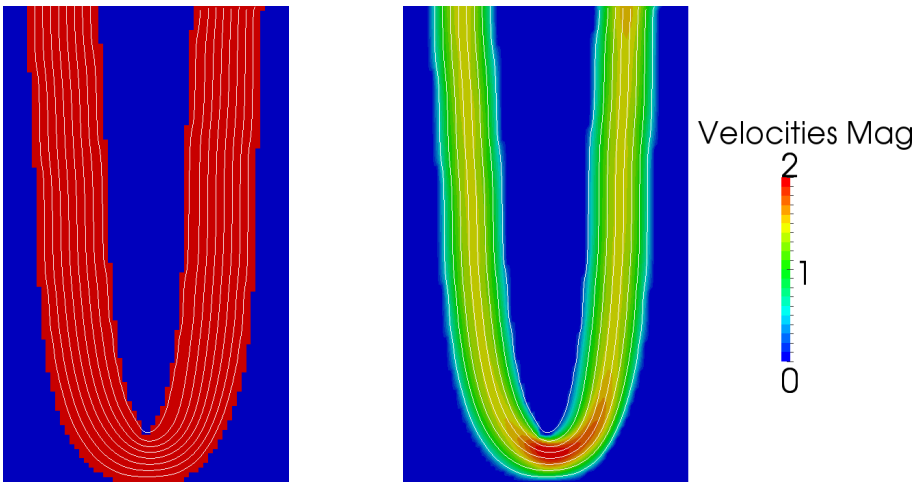
The U-bend problem, as shown in figure H.17, with $B_m 0.02$ is now optimised with respect to minimum dissipated energy, as described in appendix G.2.1. The design domain is the entire calculation domain except for the middle bar, which is kept at maximum Brinkman penalisation.

Figure H.17 shows the optimised results for the U-bend problem at $Re = 1$ and $Re = 100$ using $\overline{\alpha} = 10^6$ and $\overline{\alpha} = 10^4$, respectively. It can be observed that the designs are almost identical, which is also the case for the equivalent for $Re = 10$ (not shown). This is similar to observed in [36] where the problem is optimised using the levelset method combined with X-FEM. Here the designs for $Re = 1$ and $Re = 10$ are also more or less identical. The above results also perform almost identically in a cross-check to within 1 percent of eachother for both situations. The optimised design show that the Brinkman approach is applicable to the U-bend problem, when the design process is not unneccesarily inhibited as in [36]. In [36], the design is inhibit from placing material in a thin region around the middle bar which causes the Brinkman approach to fail.

Figure H.18 shows the velocites across the midline of the fixed porous bar for the optimised designs in figure H.17 compared to the equaivalent velocities for the empty design domain, as in figure H.1. It can be seen that the optimised design have decreased the magnitudes of the velocities considerably as expected. This highlights that the Brinkman approach can inhibit flow through thin regions by padding it with extra material, if the design domain allows it.



(a) $Re = 1, \overline{\alpha} = 10^6$



(b) $Re = 100, \overline{\alpha} = 10^4$

Figure H.17 – Design and velocity fields for the U-bend problem at $Re = 1$ and $Pe = 100$.

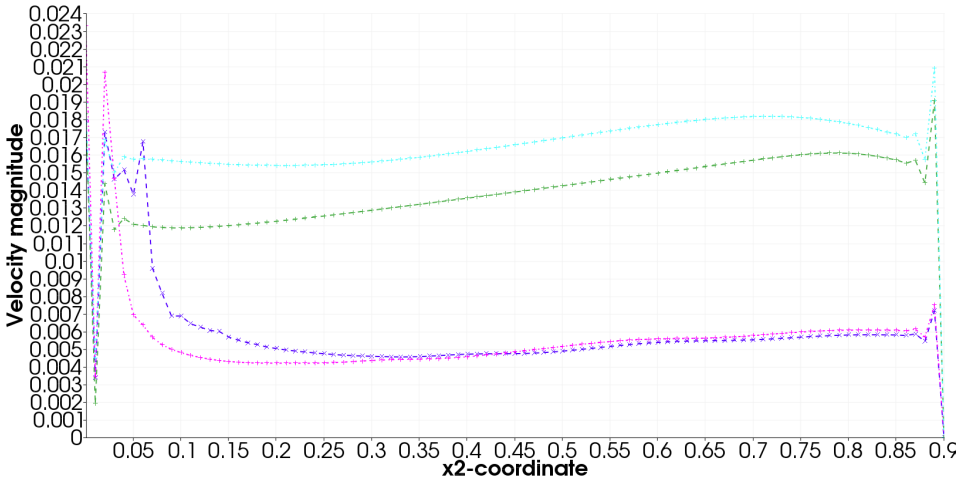


Figure H.18 – Velocities across the midline of the fixed porous bar for the optimised designs in figure H.17 compared to the equaivalent velocities for the empty design domain, as in figure H.1. In lack of a legend: purple dashed line is $Re = 100$ design, pink dash-dotted line is $Re = 1$ design, blue dash-dotted line is $Re = 100$ no design, green dashed line is $Re = 1$ no design.

Bibliography

[1] Website: Thermal-fluids-central, December 2012. URL `http://www.thermalfluidscentral.org/encyclopedia/index.php/Thermophysical_Properties:_Engine_Oil,_Unused`.

[2] Website: Engineering toolbox, December 2012. URL `http://www.engineeringtoolbox.com/thermal-conductivity-d_429.html`.

[3] Website: Wikipedia, December 2012. URL `http://en.wikipedia.org/wiki/List_of_thermal_conductivities`.

[4] N. Aage and B. S. Lazarov. Parallel framework for topology optimization using the method of moving asymptotes. *Structural and Multidisciplinary Optimization*, Published online, 2013.

[5] J. Alexandersen. Topology Optimisation of Convection Problems - B.Eng. thesis - Technical University of Denmark, 2011.

[6] J. Alexandersen. Topology Optimisation of Axisymmetric Convection Problems - Project report - Technical University of Denmark, 2011.

[7] J. Alexandersen. Finite element modelling of coupled convection problems. Technical report, Technical University of Denmark, 2012.

[8] P. R. Amestoy, I. S. Duff, and J. Y. L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Computer Methods in Applied Mechanics and Engineering*, 184:501–520, 2000.

[9] C. S. Andreasen, A. R. Gersborg, and O. Sigmund. Topology optimization of microfluidic mixers. *International Journal for Numerical Methods in Fluids*, 61:498–513, 2009.

[10] Y. Bazilevs, V. M. Calo, T. E. Tezduyar, and T. J. R. Hughes. Yzβ discontinuity capturing for advection-dominated processes with application to arterial drug delivery. *International Journal for Numerical Methods in Fluids*, 54:593–608, 2007.

[11] M. P. Bendsøe. Optimal shape design as a material distribution problem. *Structural optimization*, 1:193–202, 1989. ISSN 0934-4373.

[12] M. P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71:197–224, 1988.

[13] M. P. Bendsøe and O. Sigmund. *Topology Optimization: Theory, Methods and Applications*. Springer, 2003. ISBN: 3-540-42992-1.

[14] T. Borrvall and J. Petersson. Topology optimization of fluids in Stokes flow. *International Journal for Numerical Methods in Fluids*, 41:77–107, 2003.

[15] M. Braack, E. Burman, V. John, and G. Lube. Stabilized finite element methods for the generalized oseen problem. *Computer Methods in Applied Mechanics and Engineering*, 196:853–866, 2007.

[16] A. N. Brooks and T. J. Hughes. Streamline Upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32: 199–259, 1982.

[17] T. Bruns. Topology optimization of convection-dominated, steady-state heat transfer problems. *International Journal of Heat and Mass Transfer*, 50:2859–2873, 2007.

[18] V. J. Challis and J. K. Guest. Level set topology optimization of fluids in Stokes flow. *International Journal for Numerical Methods in Engineering*, 79:1284–1308, 2009.

[19] H.-S. Chen and M. A. Stadtherr. A modification of Powell's dogleg method for solving systems of nonlinear equations. *Computers and Chemical Engineering*, 5: 143–150, 1981.

[20] R. Codina. A discontinuity-capturing crosswind-dissipation for the finite element solution of the convection-diffusion equation. *Computer Methods in Applied Mechanics and Engineering*, 110:325–342, 1993.

[21] R. D. Cook, D. S. Malkus, and M. E. Plesha. *Concepts and Applications of Finite Element Analysis*. John Wiley & Sons.

[22] E. M. Dede. Multiphysics optimization, synthesis, and application of jet impingement target surfaces. In *The 12th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*, 2010.

[23] Y. Deng, Z. Liu, P. Zhang, Y. Liu, and Y. Wu. Topology optimization of unsteady incompressible Navier-Stokes flows. *Journal of Computational Physics*, 230:6688–6708, 2011.

[24] Y. Deng, Z. Liu, and Y. Wu. Topology optimization of steady and unsteady incompressible Navier-Stokes flows driven by body forces. *Structural Multidisciplinary Optimization*, Published online, 2012.

[25] P. Deuflhard. *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*. Springer Series in Computational Mathematics 35, 2011.

[26] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. John Wiley & Sons, 2003. ISBN: 0-471-49666-9.

[27] X.-B. Duan, Y.-C. Ma, and R. Zhang. Shape-topology optimization for navier-stokes problem using variational level set method. *Journal of Computational and Applied Mathematics*, 222:487–499, 2008.

[28] M. B. Duhring, J. S. Jensen, and O. Sigmund. Acoustic design by topology optimization. *Journal of Sound and Vibration*, 317:557–575, 2008.

[29] A. Gersborg-Hansen, O. Sigmund, and R. Haber. Topology optimization of channel flow problems. *Structural Multidisciplinary Optimization*, 30:181–192, 2005.

[30] M. B. Giles and N. A. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*, 65:393–415, 2000.

[31] J. K. Guest and J. H. Prevost. Topology optimization of creeping fluid flows using a Darcy-Stokes finite element. *International Journal for Numerical Methods in Engineering*, 66:461–484, 2006.

[32] T. J. Hughes, L. P. France, and M. Balestra. A new finite element formulation for computational fluid dynamics V - circumventing the Babuska-Brezzi condition: a stable Petrov-Galerkin formulation of the Stokes problem accomodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering*, 59:85–99, 1986.

[33] A. Iga, S. Nishiwaki, K. Izui, and M. Yoshimura. Topology optimization for thermal conductors considering design-dependent effects, including heat conduction and convection. *International Journal of Heat and Mass Transfer*, 52:2721–2732, 2009.

[34] K. Khadra, P. Angot, S. Parneix, and J.-P. Caltagirone. Fictitious domain approach for numerical modelling of Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 34:651–684, 2000.

[35] E. A. Kontoleontos, E. M. Papoutsis-Kiachagias, A. S. Zymaris, D. I. Papadimitriou, and K. C. Giannakoglou. Adjoint-based constrained topology optimization for viscous flows, including heat transfer. *Engineering Optimization*, pages 1–21, 2012.

[36] S. Kreissl and K. Maute. Levelset based fluid topology optimization using the extended finite element method. *Structural Multidisciplinary Optimization*, 46:311–326, 2012.

[37] S. Kreissl, G. Pingen, and K. Maute. Topology optimization for unsteady flow. *International Journal for Numerical Methods in Engineering*, 2011.

[38] K. Lee. *Topology Optimization of Convective Cooling System Designs*. PhD thesis, University of Michigan, 2012.

[39] A. Masud. A stabilized mixed finite element method for darcy-stokes flow. *International Journal for Numerical Methods in Fluids*, 54:665–681, 2007.

[40] T. Matsumori, A. Kawamoto, and T. Kondoh. Topology optimization for fluid-thermal interaction problems. In *6th China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems*, June 2010.

[41] C. McConnell and G. Pingen. Multi-layer, pseudo 3D thermal topology optimization of heat sinks. In *Proceedings of the ASME 2012 International Mechanical Engineering Congress & Exposition*, Houston, Texas, USA, November 9-15 2012. IMECE.

[42] J. Nocedal and S. J. Wright. *Numerical Optimization, Second Edition*. Springer Series in Operation Research and Financial Engineering. Springer, 2006. ISBN: 0-387-30303-0.

[43] F. Okkels and H. Bruus. Scaling behavior of optimally structured catalytic microfluidic reactors. *Phys. Rev. E*, 75:016301, Jan 2007.

[44] L. H. Olesen, F. Okkels, and H. Bruus. A high-level programming-language implementation of topology optmization applied to steady-state Navier-Stokes flow. *International Journal for Numerical Methods in Engineering*, 65:957–1001, 2006.

[45] C. Othmer. A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows. *International Journal for Numerical Methods in Fluids*, 58:861–877, 2008.

[46] G. I. N. Rozvany. Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics. *Structural and Multidisciplinary Optimization*, 21:90–108, 2001.

[47] G. I. N. Rozvany, M. Zhou, and T. Birker. Generalized shape optimization without homogenization. *Structural Optimization*, 4:250–252, 1992.

[48] J. H. Seo. *Optimal Design of Material Microstructure for Convective Heat Transfer in a Solid-Fluid Mixture*. PhD thesis, University of Michigan, 2009.

[49] O. Sigmund. Morphology-based black and white filters for topology optimization. *Structural Multidisciplinary Optimization*, 33:401–424, 2007.

[50] M. Stolpe and K. Svanberg. On the trajectories of penalization methods for topology optimization. *Structural Multidisciplinary Optimization*, 21:128–139, 2001.

[51] M. Stolpe and K. Svanberg. An alternative interpolation scheme for minimum compliance topology optimization. *Structural Multidisciplinary Optimization*, 22:116–124, 2001.

[52] B. Stroustrup. *The C++ Programming Language*. Addison-Wesley.

[53] K. Svanberg. The method of moving asymptotes - a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24:359–373, 1987.

[54] K. Svanberg. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal of Optimization*, 12:555–573, 2002.

[55] K. Svanberg. MMA and GCMMA, versions September 2007. Technical report, Optimization and Systems Theory - KTH Stockholm, Sweden., 2007.

[56] C. Taylor and P. Hood. A numerical solution of the Navier-Stokes equations using the finite element technique. *Computers and Fluids*, 1:73–100, 1973.

[57] C. Taylor and T. G. Hughes. *Finite Element Programming of the Navier Stokes Equations*. Pineridge Press, 1981.

[58] T. E. Tezduyar. Finite element stabilization parameters computed from element matrices and vectors. *Computer Methods in Applied Mechanics and Engineering*, 190:411–430, 2000.

[59] T. E. Tezduyar. Computation of moving boundaries and interfaces and stabilization parameters. *International Journal for Numerical Methods in Fluids*, 43:555–575, 2003.

[60] T. E. Tezduyar. Finite element methods for fluid dynamics with moving boundaries and interfaces. *Encyclopedia of Computational Mechanics*, Volume 3 Fluids:545–577, 2004.

[61] T. E. Tezduyar and Y. J. Park. Discontinuity-capturing finite element formulations for nonlinear convection-diffusion-reaction equations. *Computer Methods in Applied Mechanics and Engineering*, 59:307–325, 1986.

[62] T. E. Tezduyar, S. Mittal, S. Ray, and R. Shih. Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. *Computer Methods in Applied Mechanics and Engineering*, 95:221–242, 1992.

[63] T. E. Tezduyar, S. Ramakrishnan, and S. Sathe. Stabilized formulations for incompressible flows with thermal coupling. *International Journal for Numerical Methods in Fluids*, 57:1189–1209, 2008.

[64] F. Wang, J. S. Jensen, and O. Sigmund. Robust topology optmization of photonic crystal waveguides with tailored dispersion properties. *J. Opt. Soc. Am. B*, 28:3: 387–397, 2011.

[65] F. M. White. *Viscous fluid flow - 3rd International Edition.* McGraw-Hill, 2006. ISBN: 006-124493-X.

[66] G. H. Yoon. Topological design of heat dissipating structure with forced convective heat transfer. *Journal of Mechanical Science and Technology*, 24:1225–1233, 2010.

[67] G. H. Yoon. Topology optimization for stationary fluid-structure interaction problems using a new monolithic formulation. *International Journal for Numerical Methods in Engineering*, 82:591–616, 2010.

[68] M. Zhou and G. I. N. Rozvany. The COC algorithm, part II: Topological, geometrical and generalized shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 89:309–336, 1991.

[69] S. Zhou and Q. Li. A variational level set method for the topology optimization of steady-state Navier-Stokes flow. *Journal of Computational Physics*, 227:10178–10195, 2008.

[70] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method for Fluid Dynamics - Sixth Edition.* Butterworth-Heinemann, 2005. ISBN: 0-7506-6322-7.